PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 19/00, 17/30

A1

(11) International Publication Number: **WO 99/26180**

(43) International Publication Date: 27 May 1999 (27.05.99)

(21) International Application Number: PCT/US98/24339

(22) International Filing Date: 13 November 1998 (13.11.98)

(30) Priority Data:
08/970,359 14 November 1997 (14.11.97) US

(63) Related by Continuation (CON) or Continuation-in-Part (CIP) to Earlier Application
US 08/970,359 (CIP)
Filed on 14 November 1997 (14.11.97)

(71) Applicant *(for all designated States except US)*: NEW YORK UNIVERSITY [US/US]; 70 Washington Square South, New York, NY 10012–1091 (US).

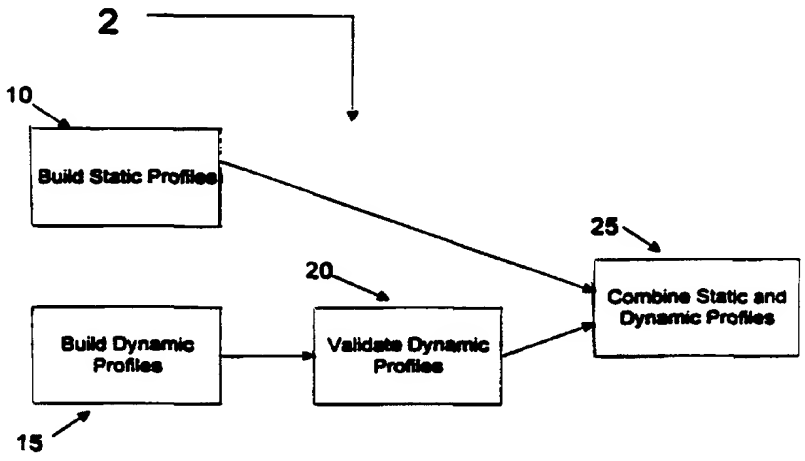
(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: TUZHILIN, Alexander [US/US]; Apartment 17B, 110 Bleeker Street, New York, NY 10012 (US). ADOMAVICIUS, Gediminas [LT/US]; 624 Newark Avenue, Jersey City, NJ 07306 (US).

(74) Agents: MEAGHER, Thomas, F. et al.; Kenyon & Kenyon, One Broadway, New York, NY 10004 (US).

(81) Designated States: CA, IL, JP, US, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: SYSTEM AND METHOD FOR DYNAMIC PROFILING OF USERS IN ONE–TO–ONE APPLICATIONS AND FOR VALIDATING USER RULES



(57) Abstract

A system and method for generating and validating a user profile (25) for a user based on a static profile (10) and a dynamic profile (15) of the user. The method and system compresses the dynamic rules (15) into aggregated rules so that the user can view a comparatively small number of the aggregated rules and select the desired rules from the aggregated rules based on user–desired criteria. The method and system validates user rules (60) using a processing device, which are retrieved from a storage device. The user rules are seperated into at least one subset of a user set. Then, it is determined if a particular rule of the at least one subset is one of acceptable, unacceptable and undecided based on a defined criteria (415). If the particular rules of the at least one subset are acceptable, the particular rules of the at least one subset are provided (e.g. assigned) to a corresponding user (435).

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

SYSTEM AND METHOD FOR DYNAMIC PROFILING OF USERS IN ONE-TO-ONE APPLICATIONS AND FOR VALIDATING USER RULES

FIELD OF THE INVENTION

The present invention relates to a system and method for dynamic profiling of a user in one-to-one marketing applications.

BACKGROUND INFORMATION

Many organizations collect historical data about every transaction that every customer performs with that organization. Such historical transactional data is useful in various one-to-one marketing applications, such as, e.g., shopping assistant application and dynamic Web site content presentation. A number of problems have been encountered in these marketing applications. One such problem relates to the creation of highly pertinent and comprehensible individual user profiles that are derived from the historical transactional data. In addition, it is also important to have the ability to utilize these user profiles when the marketing application obtains a current status of the user. If the user profiles are generated in a highly relevant and comprehensible manner with respect to a specific user, the applications would be able to understand that user's needs better and more efficiently serve that user.

There are two basic types of user profiles that can be generated - a "static" profile and a "dynamic" profile. The static profile contains all of the factual information of the user including, for example, demographic data (e.g., age, sex, address), psychographic data (e.g., personality traits and habits), purchasing preferences (e.g., what does the user purchase in an average week), etc. Static profiles are generated using conventional methods that are known to those of ordinary skill in the art.

Dynamic profiling information includes specific rules describing the user's behavior. For example, such rules may include: "whenever user X travels to France, user X often buys expensive wines in Paris" or "when user Y shops on a

weekend and did not buy any groceries for at least 3 days, user Y usually purchases a large amount of groceries." These rules can be generated with transactional data for each user using various rule generation methods that are generally known to those of ordinary skill in the art. For example, one such conventional rule generation method is implemented in a rule learning system which generates behavior rules for individual customers. (See T. Fawcett et al., "Combining Data Mining and Machine Learning for Effective User Profiling", Proceedings of the KDD'96 Conference, 1996, pp. 8-13).

In order to obtain an extensive understanding of the user, it is desirable to build both static and dynamic profiles for that user. Although the generation of static profiles is generally straight-forward, generating dynamic profiles for a large number of users may present potential problems. Many transactional systems (e.g., airline reservations systems, credit card transactional systems and/or Web site management systems) generate a various number of transactions for each user. For example, some systems and/or applications may only generate a dozen transactions per each user, which may not be enough to construct a statistically significant and reliable set of rules for a specific user. Even if there are enough transactions to construct a statistically significant set of rules, these rules should still be verified for their pertinence to the user. Since there can be a large number of users, and since the rules generated for each user may not be reliable, there is a problem of verifying a large set of generated rules for the users. For example, in a typical system facilitating 5 million users and providing about 100 rules per user, approximately 500 million rules would have to be either stored or processed. Generally, many of these rules are either not useful or insignificant. Thus, due to the amount of these generated rules, a rule validation process becomes considerably complicated. Furthermore, checking the usefulness of these rules "by hand" becomes practically impossible.

Conventional systems have not successfully provided detailed solutions to constructing reliable dynamic profiles for the users. One such system (described in T. Fawcett et al., "Combining Data Mining and Machine Learning for Effective User Profiling", Proceedings of the KDD'96 Conference, 1996) provides a limited generation of user's dynamic profiles. However, this conventional system does not provide a comprehensive method and system for analyzing a large number of dynamic rules, and thus does not provide adequate assistance for the user.

SUMMARY OF THE INVENTION

The system and method according to the present invention generates dynamic profiles and, thereafter, transforms the dynamic profiles for various users into aggregate rules. In particular, "similar" individual rules are compressed into a smaller number of aggregated rules. Because the total number of aggregate rules is substantially smaller than the total number of individual rules for all of the users, the aggregate rules can be examined manually by a human expert. This expert examines these aggregated rules and selects only rules based on the expert's preferences. Only the individual rules that correspond to the aggregated rules selected by the human expert are retained in the user's profiles. Since the selected aggregate rules were selected by the human expert, a creation of more accurate dynamic profiles is further assured. The system and method according to the present invention thus provide a more useful set of individual rules for each user.

The dynamic profiles generated with the system and method according to the present invention can be used in various systems (e.g., Personal Shopping Assistant and Personal Intelligent Digital Assistant) to provide better recommendations to the users as to which products and services each individual user should utilize. Accordingly, the user would be more satisfied with these systems and the suggestions that these systems provide to the user. In addition, Dynamic

Web Content Presentation systems can include the system and method according to the present invention because the users will be provided with better quality profiles to facilitate the provision of more pertinent Web pages to the user visiting a particular Web site. Fraud detection systems may also include the system and method according to the present invention, thus providing higher quality user profiles which may facilitate better fraud detection. Other applications for the system and method according to the present invention are also conceivable to those of ordinary skill in the art.

In addition, the system and method according to the present invention utilizing the above-described rule compression method is not limited to a construction of pertinent dynamic profiles, and can be used in a vast variety of applications (e.g., construction of high quality association rules in data mining applications). Other usages of the system and method according to the present invention are also conceivable to one having ordinary skill in the art.

In another embodiment of the system and method according to the present invention, the user rules are validated using a processing device. These user rules are retrieved from a storage device. The user rules are then separated into at least one subset of a user set. Then, it is determined if particular rules of the at least one subset is one of acceptable, unacceptable and undecided based on a defined criteria. If the particular rules of at least one subset are acceptable, the particular rules of the at least one subset are provided to a corresponding user.

When constructing good dynamic profiles, validation is an important consideration. It is possible to construct dynamic profiles for individual customers using any existing data mining methods. For example, an exemplary user rule may indicate that whenever a user buys milk in the evening, this user also buys onions. It is difficult to ascertain if this rule adequately describe the user's behavior. In fact, it may be a statistical coincidence. Therefore, it is preferable to

allow the user (or a human expert) to examine groups of the user rules.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a top level diagram of a process for generating user profiles.

Fig. 2 shows a flow diagram for generating static and dynamic user profiles.

Fig. 3 shows a flow diagram of a process for compressing dynamic rules, generating aggregate rules, validating the aggregate rules and creating user profiles.

Fig. 4 shows a detailed flow diagram of an exemplary rule compression process according to the present invention.

Fig. 5 shows a detailed flow diagram of an exemplary cluster compression process according to the present invention.

Fig. 6a shows an exemplary system for generating user profiles according to the present invention.

Fig. 6b shows a first system for generating the user profiles according to the present invention as illustrated in Fig. 6a.

Fig. 6c shows a second system for generating the user profiles according to the present invention as illustrated in Fig. 6a.

Fig. 7 shows a block diagram of an exemplary Personal Intelligent Digital Assistant system according to the present invention.

Fig. 8 shows a flow diagram of another embodiment of the process according to the present invention in which individual user rules are selectively validated using a selective validation module.

Fig. 9 shows a flow diagram of an exemplary embodiment of a process executed by the selective validation module (illustrated in Fig.8).

Fig. 10 shows a flow diagram of an exemplary procedure to generate the attribute hierarchy and to provide a cluster operation.

Fig. 11 shows a detailed illustration of an exemplary procedure to generate "Cut" data.

Fig. 12 shows an exemplary procedure for grouping subsets using the "Cut" data.

Fig. 13 shows an exemplary attribute hierarchy which can be utilized with this embodiment of the process and system according to the present invention.

Fig. 14 shows an exemplary illustration of a first level extension and a second level extensions of an exemplary node/group illustrated in Fig. 13.

Fig. 15 shows an exemplary implementation of the process and system according to this embodiment of the present invention as illustrated in Figs. 8 and 9.

DETAILED DESCRIPTION OF THE INVENTION

In many customer-related applications (e.g., banking, credit card, Internet marketing applications, etc.), user profiles for each user (or customer) are generated to better understand the user (i.e., user's purchasing trends, business travel locations, types of favorite restaurants, etc.). A flow diagram of an exemplary process for building user profiles is illustrated in Fig. 1. In particular, information regarding, e.g., the user's past purchasing history is retrieved in step 1. In step 2, user profiles are built, and the process completion is signaled in step 3. User profiles can preferably be generated using static profiles and dynamic profiles. A more detailed flow diagram of the process of building user profiles (represented in Fig. 1 by step 2) is illustrated in Fig. 2. The static profile includes user static characteristics (e.g., name of the user, address, telephone number, date of birth, sex, income, etc.). The static profile is built in step 10 using methods known to one having ordinary skill in the art. After the static profile is built, this static profile is stored in a separate file based on the data obtained from the CUST and TRANS files, as discussed below. The "CUST" file has the following format:

CUST(Cust_ID, A_1, A_2 ... A_m),

where Cust_ID is a user identifier that provides an index value for locating a specific user in the CUST file. A_1, A_2 ... A_m are fields describing the characteristics of the user (e.g., sex, income, education, etc.).

The dynamic profile is built in step 15. A dynamic profile consists of rules (or patterns) characterizing a user's behavior, e.g., "if user X shops in the evening on weekdays and purchases diapers, user X also buys beer", "if user X shops on weekdays, user X usually buys a small number of items", "if user X travels to New York on business, user X prefers to have lunches at expensive seafood restaurants." The rules are derived from a set of transactions pertaining to a particular user. These transactions may be, for example, credit card transactions, airline reservations and Web site visit transactions, and are stored in the "TRANS" file which has the following format:

TRANS(Trans_ID, Cust_ID, C_1, C_2, ... C_n)

where Trans_ID corresponds to a unique index key that identifies the transaction being performed by the user. Fields C_1, C_2, ... C_n identify a particular transaction (e.g., date of transaction, time of transaction, amount spent, location of the transaction, etc.). The field "Cust_ID" corresponds to an index key pointing to a particular user having a respective record in the CUST file. Thus, the user performing a particular transaction can be identified. Other file formats can also be utilized, as can be understood by those having ordinary skill in the art. For example, the user-specific information can also be stored in several files rather than in a single CUST file (thus, the CUST file can be normalized by splitting the CUST file into several smaller files). Using different file formats does not affect the operability of the system and process according to the present

invention. After the dynamic profile for a particular user is generated, this dynamic profile is validated in step 20.

After the validation of the dynamic profile, the static and dynamic profiles are combined to form a combined user profile in step 25. The following exemplary information can be obtained from the TRANS file to be provided into the static profile when the static and dynamic profiles (the CUST and TRANS files) are combined: a) an average transaction amount for user X; b) user X's favorite brand of beer is, e.g., Heineken; c) user X shops mostly on week-ends.

While it is relatively uncomplicated to construct user static profiles, it is much more difficult to construct quality dynamic profiles. Rules provided in the dynamic profile are generated for each user. Because a user may perform only a small number of transactions, the corresponding rules generated may be statistically insignificant, unreliable and irrelevant. In many systems (e.g, airline reservations systems, credit card transactional systems, or Web site usage systems), it is possible to have from as little as a few dozen to a few hundred transactions per each user. The rules generated with such amounts of data are often ineffective and insignificant.

The total number of generated rules can also be very large. For example, in a system serving 5 million customers and generating an average of 100 rules per user, a total number of generated rules can reach 500 million. Many of the 500 million generated rules are of questionable quality and usefulness. In order to filter the rules having such undesirable characteristics, a human expert must decide which dynamic rules should be stored and which dynamic rules should be discarded. It would be impossible for the human expert to manually check the usefulness of all 500 million rules.

Quality dynamic profiles are generated by validating dynamic rules generated using various rule induction methods. Ultimately, however, the human expert validates the machine-generated rules to determine their "usefulness" in various systems. Since most of the systems generate too many rules to

be manually examined by human experts, the system and method according to the present invention facilitates compressing individual rules into "aggregated" rules. After the individual rules are compressed into the aggregated rules, the aggregated rules are evaluated by a human expert who selects only the rules that the expert believes are pertinent for the user. In addition, it is possible (in some applications) that the respective user can be such a human expert (and examining only the rules that are pertinent to the respective user).

A. Dynamic Profile Construction Procedure

It can be assumed that user-specific rules have been already created using methods known to those having ordinary skill in the art. For example, individual user rules can be generated using an induction software system (e.g., "CART" Breiman et al., 1984; C4.5, Quinlan, 1993; or RL, Clearwater & Provost, 1990). The structure of these rules has, preferably, the following form:

$$C_{i1}\theta_{i1}a_{i1} \wedge C_{i2}\theta_{i2}a_{i2} \wedge \ldots \wedge C_{ik}\theta_{ik}a_{ik} \Rightarrow C_i\theta_i a_i \qquad (1)$$

where C_{i1}, C_{i2}, ..., C_{ik}, C_i are fields from the TRANS file, a_{i1}, a_{i2}, ..., a_{ik}, a_i are constants, and θ_{ij} are relational operators (e.g., "=", ">", "<", etc.). In addition, each rule is assigned to a user defined by the Cust_ID (user identifier) from the CUST file.

Next, it is important to remove "useless" individual rules from the total number of rules. A process to remove these useless individual rules is shown in Fig. 3. In step 30, individual rules are provided for processing. In step 35 several "similar" individual rules (of the form (1)) are compressed into one aggregated rule of the form:

$$A_{i1}\theta_{i1}b_{i1} \wedge A_{i2}\theta_{i2}b_{i2} \wedge \ldots \wedge A_{ij}\theta_{ij}b_{ij} \wedge C_{i1}\theta_{i1}a_{i1} \wedge C_{i2}\theta_{i2}a_{i2} \wedge \ldots \wedge C_{ik}\theta_{ik}a_{ik} \Rightarrow C_i\theta_i a_i \qquad (2)$$

where A_{i1}, ..., A_{ij} are the fields in the CUST file, b_{i1}, ..., b_{ij} are constants, and θ_{ij} are relational operators (e.g., "=", ">", "<", etc.). For each individual rule of the form (1), the aggregated rule of the form (2) is formed after the individual rules are compressed. The newly aggregated rules (formed in step 40) can be, e.g., fuzzy rules, and the operators θ_{ij} should also be, e.g., fuzzy operators. For example, several of the individual rules that are similar (generally pertaining to different users) can be compressed into one aggregated rule pertaining to the same subject matter that can be applicable to several users. For example, if several rules have the form:

IF Shopping_time = "evening" and Day_of_week = "weekday" and Purchase = "diapers" THEN Purchase = "beer",

and it is known that most of the users corresponding to this rule are males, then these rules can be compressed into the aggregated rule having the following form:

IF Sex = "Male" and Shopping_time = "evening" and Day_of_week = "weekday" and Purchase = "diapers" THEN Purchase = "beer".

Additional fields (e.g., Sex, etc.), unlike other fields in the above exemplary rule, are fields from the CUST file. Individual rules relating to different users can be compressed into the same aggregated rule for a group of users. Thus, the rule compression can preferably be implemented for different users. The number of aggregated rules (of the form (2)) generated by the compression algorithm should be much smaller than the initial number of individual rules. Then, in step 45, the aggregated rules can be validated (one by one) by the human expert (including a particular user) to determine which rules are appropriate for that user. After the user validates the aggregated rules, this user selects the set of preferred aggregated rules in step 50. Only the individual rules corresponding to the aggregated rules selected in step

50 are retained in the user's dynamic profile (step 55) to provide validated individual rules (step 60) to the user.

B. Rule Compression Process

Fig. 4 illustrates a detailed flow diagram of an exemplary rule compression process (starting from step 35 in Fig. 3). Two individual rules of the form (1) are referred to as "similar" rules if they differ from each other only in the values of their respective constants a_{ij}. Thus, similar rules should have the same number of terms, the same fields C_{ij}, and the same comparison operators θ_{ij}. Similar rules can be mapped into the (k+1) dimensional space defined by $Dom(C_{i1}) \times \ldots \times Dom(C_{ik}) \times Dom(C_i)$, where Dom is a domain (or range of values) of the field C, with a rule having the form (1) being mapped into the points (i.e., a_{i1}, a_{i2}, ..., a_{ik}, a_i). This set of points is generated by similar rules. For example, the rule "if user X shops in the evening on weekdays and purchases diapers, user X also buys beer" can be written as:

IF (Shopping_time = "evening" and Day_of_week = "weekday" and Purchase = "diapers") THEN Purchase = "beer".

This sample rule would be mapped into the corresponding vector ("evening", "weekday", "diapers", "beer") of the 4-dimensional space of attributes (variables):Shopping_time, Day_of_week, Purchase and another Purchase.

The exemplary rule compression process (described below in detail) then generates rules (e.g., fuzzy rules of the form (2)). These fuzzy rules utilize fuzzy linguistic variables for the fields from the CUST and TRANS files, which are generally known to those having ordinary skill in the art. Each fuzzy linguistic variable has a corresponding identifier (e.g., Income, Transaction_Amount, etc.), each being capable of providing a range of values (e.g., natural numbers between 0 and 1,000,000), a set of terms (e.g., "low", "medium", "high", etc.), and a membership function that assigns a membership value (e.g., between 0 and 1) to each value from the domain of the fuzzy linguistic variable for each range of

values. In addition, the non-ordered fields in the CUST and TRANS files (e.g., "Product_Purchased") have assigned classification hierarchies; for example, the field "Product_Purchased" can include standard classification hierarchies used in marketing. Thus, UPCs, e.g., can be grouped into brands, brands can be grouped into product categories, etc.

The following exemplary inputs are provided to the Rule Compression Process:

a. Individual rules from users' dynamic profiles.

b. Fuzzy linguistic variables for all fields in the CUST and TRANS files.

c. Hierarchical classifications for non-ordered fields.

Exemplary outputs generated by the Rule Compression Process are a set of (preferably) fuzzy aggregated rules having the form (2).

The steps of the exemplary rule compression process shall now be described in detail with reference to Fig. 4. In step 160, all the individual rules of the form (1) are grouped into sets of similar rules (i.e., rules having the same structure). The maximal number of such similar groups is 4^n, where n is the number of fields in the TRANS file. For example, if n = 10, then there can be at most 1×2^{20} similar groups. However, this number is typically much smaller in practice. Each set of similar rules forms a set of points in a k-dimensional space generated by the individual rules described above. In step 165, a group of clusters of the generated points is determined using any of the cluster computation methods known to those of ordinary skill in the art. In step 170, starting from the first cluster of the group of clusters determined in Step 165, an approximate rule for that cluster is determined in Step 180. The approximate rule is determined as a function of the points in the cluster.

For example, a point in the cluster may be the "center" of the cluster. The center can be identified as a point that minimizes the sum of distances from a particular point to other points in the cluster. For example, given

Cluster $C_i = (c_{i1}, c_{i2}, \ldots c_{ik})$, the center of this cluster is the point that minimizes the expression:

$$\min_{c_i} \sum_{x \in Clust_i} d(x, c_i)$$

The center of the cluster can also be determined using other methods, such as, e.g., selecting the most "representative" point in that cluster.

In step 185, the next cluster is selected and the procedure described with respect to step 180 is repeated. In step 175, it is determined whether all of the clusters in the group of clusters have been evaluated. As an illustration, if cluster $Clust_i$ contains three 3-dimensional points (0,0,1), (0,1,0), (1,0,1), corresponding to the vertices of a equilateral triangle, then the center of this cluster C_i is the center of the triangle, i.e., the point (½,½,½). Other approaches to defining the center of a cluster can be used. This exemplary rule compression process does not depend on any specific method for defining any center of a cluster C_i.

Given the set of rules (1) corresponding to the cluster with the center $C_i = (c_{i1}, c_{i2}, \ldots c_{ik})$, the corresponding aggregated rule has the form:

$$C_{i1}\theta_{i1}c_{i1} \wedge C_{i2}\theta_{i2}c_{i2} \wedge \ldots \wedge C_{ik}\theta_{ik}c_{ik} \Rightarrow C_i\theta_i c_i \quad (3)$$

which is a "representative" rule for the cluster of similar rules. For example, if the center of the cluster is (½,½,½), the following rule is generated:

$$C_1 = ½ \wedge C_2 = ½ \Rightarrow C_3 = ½.$$

Also, for totally ordered fields C_{ij}, standard deviations σ_{ij} of the points in that cluster are calculated for that field. For

unordered categorical fields C_{ij}, a measure of the "deviation" of the points is determined in that cluster along the j-th dimension from c_{ij} (by using the hierarchical classification for that field).

In step 190, a total number of clusters generated in Step 165 is provided to the user. In step 195, the user is asked if there are too many of the generated clusters for manually inspecting the aggregated rules (i.e., the number of generated clusters is greater than a predetermined number). If so, the generated clusters are compressed using a cluster compression process described in step 205. Thereafter, there is a smaller number of clusters (and corresponding aggregation rules per cluster). The user is asked again, in step 195, if there are too many generated clusters for the manual inspection of aggregated rules. If the number of clusters is smaller than the predetermined number, for each cluster C_i obtained in step 165 or in step 205, a set of users corresponding to the points for that cluster is identified in step 210. Each point in a cluster corresponds to a first representative rule from the dynamic profile of the user, so that all of the users corresponding to the dynamic profile rules from that cluster can be identified. For example, $CUST\_ID_i$ is defined as a set of values $Cust\_ID_{ij}$ corresponding to the users corresponding to the rules of cluster C_i. A set of records ("$CUST_i$") from the CUST file corresponding to the users of that cluster is identified (i.e., having user ID values from the set $CUST\_ID_i$). Thus, $CUST_i = \{ r \mid CUST(r)$ and $r.Cust\_ID \in CUST\_ID_i \}$.

The set of records $CUST_i$ form a set of points in m-dimensional space (where m is the number of fields in the CUST file). These points are separated into clusters using the same techniques as described in step 165. For each resulting cluster $CUST_{ij}$, a center is located as explained below. The set of points belonging to that cluster is approximated with a logical statement having the form:

$$A_1\theta_{ij1}b_{ij1} \wedge A_2\theta_{ij2}b_{ij2} \wedge \ldots \wedge A_m\theta_{ijm}b_{ijm} \quad (4)$$

to form a corresponding condition in step 215, where A_i are the fields of the CUST file, θ_{ijl} are relational operators (e.g., "=", "<", ">", etc.) and b_{ijl} are constants. One way to construct the condition (4) would be by finding the center $b_{ij} = (b_{ij1}, \ldots, b_{ijm})$ of the cluster $CUST_{ij}$ as described in step 180, and substituting the values of b_{ijl} into the condition (4) (also setting all the relational operators to be "="). Another way to construct this condition (4) is described in A. Motro, "Using Integrity Constraints to Provide Intentional Answers to Relational Queries", Proceedings of the 15th International Conference on Very Large Databases, 1989, pp. 237-246, and C. Shum et al., "Implicit Representation of Extensional Answers", Proceedings of the 2nd International Conference on Expert Database Systems, 1988.

In step 220, the first and second representative rules are augmented (i.e., expression (4) is augmented with expression (3)). The resulting rule is:

$$A_1\theta_{ij1}b_{ij1} \wedge A_2\theta_{ij2}b_{ij2} \wedge \ldots \wedge A_m\theta_{ijm}b_{ijm} \wedge C_{i1}\theta_{i1}c_{i1} \wedge C_{i2}\theta_{i2}c_{i2} \wedge \ldots \wedge C_{ik}\theta_{ik}c_{ik} \Rightarrow C_{im}\theta_{im}c_i. \quad (5)$$

For example, assume that the center of a cluster is a rule: "if a user shops in the evening on weekdays and buys diapers, the user also buys beer" (i.e., IF Shopping_time = "evening" and Day_of_week = "weekday" and Purchase = "diapers" THEN Purchase = "beer"). Also, assume that most of the users in that cluster are men, thus forming the expression (4) where "Sex" = "Male". Accordingly, the augmented rule is "if a male user shops in the evening on weekdays and buys diapers, the user also buys beer" (i.e., IF "Sex" = "Male" and "Shopping_time" = "evening" and "Day_of_week" = "weekday" and "Purchase" = "diapers" THEN "Purchase" = "beer").

Then, in step 225, the rules of the form (5) generated in step 220 are converted into fuzzy aggregated rules. In particular, each field A_i and C_{ij} in the form (5) is mapped into a corresponding fuzzy linguistic variable associated with that field. In addition, all of the terms in

the expression (5) are converted into appropriate fuzzy expressions. For example, assume that a non-fuzzy term $A_1 = 20$ corresponds to a fuzzy linguistic variable also denoted as A_1. Further assume that the term set for A_1 is either low or high, and that there is a membership function that assigns the membership value (e.g., between 0 and 1) to each value from the domain of fuzzy term A_1 for each value from the term set. Then, it can be determined for which term (i.e., low or high) the membership value 20 is higher, and a corresponding term is assigned. If the membership value is higher for the term "low", then the expression $A_1 = 20$ is replaced by $A_1 = \text{LOW}$.

In step 230, the set of aggregated fuzzy rules generated by the rule compression process is shown to the human expert who selects only the meaningful and useful rules from this set according to user desired criteria.

C. Cluster Compression Process

Fig. 5 shows an exemplary cluster compression process as discussed above with respect to step 205 illustrated in Fig. 4. As an initial matter, it is assumed that, e.g., clusters $Clust_1$ and $Clust_2$ are determined in step 165. Since $Clust_1$ and $Clust_2$ can be generated by dissimilar rules, the rules from each of these clusters $Clust_1$ and $Clust_2$ can be very different (or similar). Therefore, it is important to determine whether two different clusters are substantially similar to each other so that they can be merged. In particular, the distance between two aggregated rules of the form (3) corresponding to the centers of these clusters is determined to ascertain whether these different clusters are substantially similar. As an example, the following two aggregated rules corresponding to the center of $Clust_1$ and $Clust_2$ are considered:

$$C_1 = a \wedge C_2 < b \Rightarrow C_4 = c, \text{ and}$$

$$C_1 = d \wedge C_3 = e \Rightarrow C_4 = g$$

It may be also assumed that the domains of attributes C_2 and C_3 are discrete and ordered. These rules have different structure and therefore are different. In order to calculate the distance between these rules, we first have to bring these rules into the same 4-dimensional space of attributes C_1, C_2, C_3, and C_4. This can be done by replacing these rules with the rules

$$C_1 = a \wedge C_2 = z \wedge C_3 = x \Rightarrow C_4 = c \quad (6)$$

$$C_1 = d \wedge C_2 = y \wedge C_3 = e \Rightarrow C_4 = g \quad (7)$$

where x and y are uniformly distributed random variables ranging over the domains $Dom(C_3)$ and $Dom(C_2)$ of attributes C_3 and C_2 respectively and *z* is a uniformly distributed random variable ranging over the domain of $Dom(C_2)$ from its smallest element to b. This procedure can also be performed using actual distribution in the data for corresponding attributes of x and y variables. It should be noted that the term $C_2 < b$ (the first aggregated rule described above) should be replaced with $C_2 = z$ in rule (6). In addition, term $C_3 = x$ is provided into the first aggregated rule and term $C_2 = y$ is provided into the second aggregated rule. It is also assumed that, e.g., random variables x, y, and *z* are uniformly distributed over their respective domains.

If constants are substituted for the variables x, y, and *z*, the terms of the aggregated rules (6) and (7) will contain only equalities and constants. Thus, these aggregated rules (with the above-described substitutions) will have respective points in the same 4-dimensional space. If the distance between these two points can be calculated for fixed values of variables x, y, and z -- $d(Clust_1 (x,z), Clust_2 (y))$ (i.e., if all the attributes are numeric, then the distance can be a Euclidean distance; if some of the attributes are categorical and unordered, the distance can be calculated in terms of how far the nodes are in the aggregation hierarchy

defined for that attribute) -- then the distance between clusters $Clust_1$ and $Clust_2$ is equal to:

$$d(Clust_1, Clust_2) = \frac{1}{Dom(C_2) * Dom(C_3) * (b - \min(Dom(C_2))} * \sum_{x \in Dom(C_3), y \in Dom(C_2), z < b} d(\quad Clust_1(x, z), Clust_2(y))$$

since it can be assumed that the domains of attributes C_2 and C_3 are discrete. If these domains were continuous, integration would have been used instead.

In general, let $c_1 = (c_{11}, c_{12}, \ldots c_{1k})$ and $c_2 = (c_{21}, c_{22}, \ldots, c_{2m})$ be the centers of two clusters $Clust_1$ and $Clust_2$ as calculated in steps 170 through 185 illustrated in Fig. 4, where c_1 and c_2 are vectors with different dimensions (because different rules can have different numbers of terms). The rules corresponding to the centers of these two clusters are extended with, e.g., dummy attributes and dummy random variables that form a union of the attributes for clusters $Clust_1$ and $Clust_2$. Assuming that the dummy variables are uniformly distributed over their domains, the distances between the two rules for fixed values of random variables can be calculated. Thereafter, the random variables are either integrated (for continuous random variables) or summed (for discrete random variable) over different values of these random variables. Thus, the distance between clusters can be determined using the system and method according to the present invention.

Once the distance between the two clusters is determined, the clusters can be merged as follows. In order to perform this operation, the size of the cluster should be determined as a part of the Cluster Compression process. The size of the cluster is the measure of how far the points of the cluster are apart from each other. This size can be determined, e.g., using the following formula:

$$size(Clust) = \frac{1}{|Clust|} \sum_{x \in Clust} d(x, c)$$

where c is the center of the cluster. Other measurements can also be used by those having ordinary skill in the art.

The flow diagram in Fig. 5 illustrates an exemplary process for compressing clusters. In particular, two clusters $Clust_1$ and $Clust_2$ are selected in step 250. There are a number of ways to determine which clusters should be selected in step 250. The simplest way to select the clusters is in an arbitrary manner. In step 260, the distance between the clusters is determined, as discussed above. In step 265, the respective size of each cluster is determined. In step 270, a check is performed to determine if the distance between the clusters $\{d(Clust_1, Clust_2)\}$ is smaller than the sizes of these clusters (e.g., to determine if these two clusters are "close enough" to each other). If so, the clusters should be merged into one cluster in step 275; otherwise, the clusters are maintained as separate clusters. In particular, an inquiry as to whether two clusters are "close enough" can be computed in the following manner, e.g.:

$$\frac{2 * d(Clust_1, Clust_2)}{size(Clust_1) + size(Clust_2)} < \alpha \qquad (8)$$

where α is a predetermined threshold value. The two clusters should be merged by forming a new cluster consisting of points from $Clust_1$ and $Clust_2$ if condition (8) occurs. Steps 250-275 should be repeated until there are no more clusters left that can be merged (see step 255).

In deciding which clusters $Clust_1$ and $Clust_2$ should be chosen in step 250 of the cluster compression process, distances between, e.g., all pairs of clusters can be calculated and condition (8) can be checked to ascertain which clusters should be merged. Other methods to select the clusters for compression can also be used. Furthermore, the

distance between all the pairs of clusters does not necessarily have to be calculated.

The system according to the present invention can be used in a Personal Shopping Assistant (PSA), a Personal Intelligent Digital Assistant (PIDA), and in a dynamic Web content presentation system, described below.

A Personal Shopping Assistant (PSA) system according to the present invention provides recommendations on the products and services that its users should consider purchasing (including, e.g., suggestions for purchasing at a specific source, and at a particular price). An exemplary embodiment of the PSA system according to the present invention is shown in Fig. 6a. In particular, the system includes a User Transaction Collection and Recording Unit (or module) 115, a Past Purchasing History Storage Unit (or module) 120, a User Profile Generation module 110, a State-of-the-World module 150, a User Estimated Purchasing Needs module 140, a Purchasing Recommendations module 145, and the State-of-the-User module 160.

The User Transaction Collection and Recording Unit 115 collects most of the shopping transactions performed by the user (e.g. 80-90% of all the purchases made by the user). The User Transaction Collection and Recording Unit 115 can be implemented as a "smart card," or as a smart Point of Sales register that records individual items purchased by the user. Alternatively, the user himself can record this information (as part of the User Transaction Collection and Recording Unit 115) using some transaction recording systems such as Quicken or Microsoft's Money.

When the user purchases one or more products, the User Transaction Collection and Recording Unit 115 records and transmits this information to the Purchasing History Storage Unit 120 where this information is stored as part of the purchasing history of the user. The Purchasing History Storage Unit 120 can be implemented, e.g., as a database that records transactions performed by various users in the TRANS file, as described above.

Information stored by the Purchasing History Storage Unit 120 is provided to User Estimated Purchasing Needs module 140. In order to estimate the user's purchasing needs, pertinent static and dynamic profiles of the user should be constructed based on the past purchasing histories obtained from the Purchasing History Storage Unit 120, which is performed by the User Profile Generation module 110. Static profiles include the user's demographic information (e.g., age, sex, marital status), particular preferences (e.g., user prefers a particular brand of beer), and certain purchasing decisions (e.g., the user bought a particular automobile in a particular month). Dynamic profiles include a set of rules (e.g., "if a user goes to France, the user often buys perfumes in Paris", "if user Y visits a Web site from the site Z in the evening, user Y does not spend a predetermined amount of time at site Z", etc.).

In addition, the PSA system maintains information on the current State of the World using the State-of-the-World module 150, which records information, e.g., on a broad range of products and services offered by various suppliers and on promotions and discounts run for these products and services. Also, the PSA system includes the State-of-the-User module 160 that maintains information about the user obtained from the Purchasing History Storage Unit 120 (e.g., the user will be in New York on June 28, 1995 because she purchased an airline ticket for that date and destination) and various external information (e.g., the date, time, and the user's location, if available).

By knowing the purchasing history of a user (provided from the Purchasing History Storage Unit 120), the user's profile (provided from the User Profile Generation module 110), and the external information about the user (provided from the State-of-the-User module 160), the PSA system estimates the user's future purchasing needs using the User Estimated Purchasing Needs module 140. This Estimated Purchasing Needs module 140 may match the rules specifying which products the user will buy (and when) with the user's

purchasing history. As a result, a set of products that the user should consider buying is produced.

Once future purchasing needs are estimated in Step 140, the PSA system will match these needs against a broad range of products and services offered by various suppliers and on the promotions and discounts run for these products and services. This matching process is performed by the Purchasing Recommendation module 145 using conventional methods that are known to those of ordinary skill in the art. For example, if the user needs to buy a pair of jeans within the next two months, the Purchasing Recommendations module 145 selects the merchants selling jeans, e.g, the cheapest pair of jeans that fits the use's requirements (considering the promotions offered within the next two months) by matching to the user profile (i.e., the user's purchasing needs). Once the Purchasing Recommendations module 145 matches the user's purchasing needs against the products and services, the Purchasing Recommendations module 145 provides purchasing recommendations to the user.

For example, based on the past purchasing history of a particular user, the PSA service may ascertain that whenever user X goes to France, user X often buys perfume in Paris. This rule is stored as a part of the user profile using the User Profile Generation module 110. In addition, the Purchasing History Storage Unit 120 of the PSA service may receive information that the user has purchased a ticket to Paris, and in a substantially same time period, the State-of-the-World Unit 150 of the PSA service also receives information that, e.g., Christian Dior has launched a new line of perfumes that is similar to the brands previously purchased by user X. In addition, the State-of-the-World Unit 150 may also receive information that the duty-free shop at Charles de Gaulle airport is having a sale on these new perfumes (the price being very competitive). Using the above-described exemplary information, the PSA service (using the User Estimated Purchasing Needs module 140) estimates that user X may want to buy these perfumes and sends a message to user X

(via the Purchasing Recommendation module 145) to consider purchasing the new perfume at the duty-free shop at Charles de Gaulle airport.

The success of the PSA service depends primarily on accurate predictions by the PSA service of users' future needs. If the user finds, e.g., 50% of the PSA suggestions useful, the user will probably be satisfied with the PSA service. However, if the user finds, e.g., only 10% of the suggestions to be useful, the user will, most likely, reject this service. As indicated above, in order to make predictions of the user's future needs more accurate, it is important to build reliable user profiles. The present invention provides a method and system for generating better dynamic profiles and, therefore, providing more accurate predictions of the users' future needs.

The PSA system illustrated in Fig. 6a can be implemented using a first exemplary system shown in Fig. 6b and a second exemplary system shown in Fig. 6c. The first exemplary system of Fig. 6b provides that the User Transaction Collection and Recording Unit 115 is stored on the client side. All other modules from Fig. 6a are stored on the server side and are connected to the User Transaction Collection and Recording Unit 115 via a Telecommunication Medium 130 (e.g., a telephone line or a wireless communication medium). In the first exemplary system, individual user purchasing histories and static and dynamic profiles of these users are stored on the server at a central location (e.g. a database), and the method and system according to the present invention (as described above) generates improved dynamic profiles, and thus provides better estimated purchasing needs of the users.

The second exemplary system of Fig. 6c provides that the User Transaction Collection and Recording Unit 115, the User's Profile Generation Module 110, the Purchasing History Storage Unit 120, the State-of-the-World module 150, the State-of-the-User module 160, and the User Estimated Purchasing Needs module 140 are stored on the client side, while the State-of-the-World module 150 and the Purchasing

Recommendations module 145 are stored on the server side. In the second exemplary system, the user dynamic profiles are validated in Step 20 of Figure 2 by the user (since these profiles are stored on the client side and are available to the user for checking and validating). Once module 140 estimates user purchasing needs, these estimated user purchasing needs are transmitted via the Telecommunication Medium 130 (e.g., a telephone line or a wireless communication medium) to the server, where the estimated user purchasing needs are matched by the Purchasing Recommendation module 145 to various products and services offered by various suppliers (that are stored on the server side). The resulting purchasing recommendations are transmitted back to the client side via the telecommunication medium 130 for the user's consideration.

The PSA service can also be used in a Personal Intelligent Digital Assistant (PIDA) service as illustrated in Fig. 7. Each user subscribing to this additional service is provided with a Personal Digital Assistant (PDA) (e.g., the remote device 350 or the User Transaction Collection and Recording Unit 115), which is connected to the PSA system (e.g., a general purpose computer 300). The PDA remote device(s) 350 (which includes, e.g., a PDA processor 360, a PDA I/O port 365 and a PDA input device 355) and the PSA system(s) 300 (which includes, e.g., a display device 310, a storage device, a PSA processor 320, a PSA/ I/O port 325 and a PSA input device 305) form a client-server architecture, in which the PDA remote device is a client and the PSA system is a server. The PSA system, using the Past Purchasing History Storage Unit 120 (e.g., a storage device 315) and the User Profile Generation module 110, the State-of-the-World module 150, the State-of-the-User module 160, the User Estimated Purchasing Needs module 140 and the Purchasing Recommendations module 145 (executed by, e.g., a processor 320) estimates users' future needs and behavior as described above. The PDA device accumulates additional information on the user's current state, such as the user's location information,

preferences, and desires (e.g., the user is hungry now and wants to eat). This additional information is transmitted from the PDA device to the PSA system via the telecommunication medium 130 (e.g., a wireless network, fiber-optics communication system, telephone wired system, etc.) to be stored using the State-of-the-User module 160 (e.g., in the storage device 315) as part of the user's state and is used subsequently for estimating the user's purchasing needs.

For example, in order to illustrate how the PIDA service operates, assume that it is Tuesday, 11:30 am and that user X is driving in his car on I-87 in the Albany region on business, and that he indicated through his PDA device 350 that he wants to have lunch. The PDA device (350) records the current state of user X as "Tuesday, 11:30 am, presently driving in user X's car on I-87 in the Albany region, travel purpose is business, wants to have lunch." This information is sent from the PDA device 350 to the PSA system 300 via telecommunication medium 130. Based on user X's past purchasing history, the PIDA service recognizes that whenever user X is traveling on business, he likes to have light lunches at good quality restaurants and that he generally likes sea food. By examining user X's personal profile, and by matching the dynamic rule which provides that "whenever user X travels on business, he prefers light lunches at good quality restaurants", with user X's current state (user X is currently traveling on business), the PSA system 300 can predict that user X prefers a lunch at a good quality restaurant and he wants to eat light food. Next, the State-of-the-world module 150 of the PSA system 300 searches for highly rated seafood restaurants in the Albany region. If the PSA system 300 finds any such restaurant, user X is provided with restaurant choices (e.g., if more than one restaurant is located) by contacting user X's PDA device 350. If the PSA system 300 does not find first choice restaurants conforming to the user X's preferences, the PSA system 300 provides second choice restaurants to user X.

User needs are estimated based on purchasing history, the user's static and dynamic profiles and the current "state" of the user (sent to the PSA system from the PDA device). When the needs of the user are estimated (e.g. the user wants to buy a perfume in Paris, or wants to eat at a good seafood restaurant in the Albany region), they are matched with the current state of the "world." If the PIDA service finds good matches (e.g., Christian Dior perfumes are on sale at Charles de Gaulle airport in Paris, or that there is a good seafood restaurant in the Albany region serving special lunches and located very close to the user's current route), purchase recommendations are provided to the customer based on these matches. These recommendations are sent back from the PSA server 300 to the PDA device 350 via a telecommunication medium 130 (e.g., via e-mail or through another intelligent user interface).

The PIDA service incorporating the system and method according to the present invention can be used for notifying the users about various purchasing opportunities, both time sensitive (e.g., a particular sale will start next week) and spatial (e.g., if you need a new sweater, and sweaters you would probably like are on sale at the store near your work).

The system and method according to the present invention can also be incorporated in a Web site system. In conventional systems, when a user visits a particular Web site, the user usually sees the same contents, regardless of who the user is. Using the system and method according to the present invention (i.e., individual profiles for respective users), the dynamic Web content of the Web site presented to the user can be varied to conform to the dynamic profile of the user visiting the Web site. Furthermore, dynamic profile construction methods can also be used in fraud detection systems. In particular, a set of fraud detection rules can be dynamically generated for each user.

It should be noted that the use of the above-described rule compression process and the cluster compression process according to the present invention is not limited to a

construction of user profiles. For example, these process can also be used for computing useful association rules in data mining applications, or in general compressing large sets of rules generated by data mining algorithms.

D. Selective Validation Procedure

Another embodiment of the present invention for providing a selective validation of individual user rules is shown in Fig. 8. In particular, user rules for all individual users (e.g., customers) are provided to a selective validation module/arrangement (step 375). The selective validation module/arrangement can be preferably executed by a central computing device illustrated in Figs. 6a and 6b, or executed by the processor 320 of the general purpose computer 300 illustrated in Fig. 7. The individual user rules may be stored in the storage device 315. It is also possible to provide the selective validation module/arrangement in the remote unit 350 illustrated in Fig. 7. In step 380, the selective validation module/arrangement receives still unvalidated user rules and outputs at least one set of selectively validated individual user rules (step 390). In addition, the selective validation module/arrangement can optionally include the process illustrated in Fig. 3. In an exemplary embodiment of the present invention, this selective validation procedure allows the human expert to select particular subsets of individual user rules and characterize these subsets as "Good" subsets, "Bad" subsets and/or "Undecided" subsets.

A flow chart representation of an exemplary embodiment of a process executed by the selective validation module (or an exemplary steps executed by the selective validation arrangement) described above is illustrated in Fig. 9. According to the present invention, a "Good_Rules" set is provided to maintain (e.g., store) all sets of individual user rules which were selected by the human expert as rules which are usable for a particular user. A "Bad_Rules" set is

provided to store all sets of individual user rules selected by the human expert to be unusable for that user.

As shown in Fig.9, (in step 400) each of the "Good_Rules" and "Bad_Rules" sets are initialized, e.g., to be empty or null sets. In step 405, all user rules are combined to form Set S. Set S initially contains all related (e.g., similar) subsets of the unvalidated individual user rules for all users. These similar subsets may be grouped in a similar manner as discussed above with reference to Fig. 4, or using a filtering and/or clustering operator as discussed below. In step 410, the user rules in Set S (or subsets in Set S) can be displayed. The human expert examines the subsets of "related" rules from Set S (e.g., one rule or one set at a time), and selects which subsets (or which rules) in Set S are "good", "bad" and/or neither (step 415). These subsets can also be examined automatically by a system (e.g., the processor 320 implementing an expert system or an artificial intelligence system) using a predetermined criteria. If a particular subset in Set S is selected to be usable, the particular subset is marked as "good"; if this subset is selected to be unusable, it is marked as "bad"; if the human expert (or the system) cannot determine if the particular subset is usable or not, such subset is marked as "undecided" (step 420). In step 425, the subsets which are marked as "good" are moved from Set S to the Good_Rules set, and the subsets which are marked as "bad" are moved from Set S to Bad_Rules set.

In step 430, a decision is made (e.g., automatically via the processor 320 or by the human expert) if the processing of the selective validation module/arrangement is completed, and, if so, initiates a completion process according to this embodiment of the present invention. There can be numerous conditions to indicate to the selective validation module/arrangement according to the present invention that the completion process should be initiated. For example, the following exemplary conditions may prompt the selective validation module/arrangement to stop processing:

- Set S can became empty (i.e., all subsets of rules are moved from Set S to "Good_Rules" set and/or to "Bad_Rules" set). If this is the case, all subsets of rules are marked with their appropriate designation (i.e., "good" or "bad");
- the number of subsets in Set S is less than a predetermined number;
- the ratio of the rules in Set S with respect to all of the existing rules is less than predetermined value; and
- the user decides to stop the process (e.g., a desired number of rules has already been classified or marked).

Other stopping criteria may be used for initiating the completion process according to the present invention.

If it is determined that the processing of the completion process according to the present invention should be initiated, the rules from the Good_Rules set is assigned to one or more corresponding users (step 435), Good_Rules set and/or undecided subsets can be displayed (step 440), and the execution of the process according to the present invention is stopped. If, however, it is determined that the completion process should not be initiated (i.e., the subsets should be regrouped), the remaining rules in Set S (i.e., the subsets marked as "undecided") are grouped or regrouped to generate a new Set S (step 445), and this new Set S is provided to the human expert (i.e., looped back to step 410) so that the rules within new Set S may be reclassified using the process and/or the arrangement according to the present invention (i.e., looped again starting with step 410).

It should be noted that if a particular subset Set S is marked as "undecided", this subset is then further analyzed by either splitting it into smaller subsets using techniques described below or optionally regrouping this particular subset with other related sets from Set S as also described below.

According to an exemplary embodiment of the process according to the present invention, the rules in Set S which were marked as "undecided" are grouped to generate a new Set S according to the following exemplary methods:

- A predetermined number of the remaining subsets (which can also be a single subset) contained in Set S are selected and merged together to form new subsets. The above-described remaining sets can be selected by the human expert or according to some predetermined selection criterion (e.g., the size of individual sets of rules should be smaller than a predetermined value).

- One or more subsets are selected from Set S. For each of these subsets, at least one of the following exemplary "partitioning" operators is applied to the selected subsets: a filtering operator and/or a cluster/grouping operator (which are as described below). Other "partitioning" operators can also be implemented. The terms - "clustering operator" and "grouping operator refer to identical operations and shall be utilized interchangeably below. In a particular embodiment of the present invention, subsets in Set S (obtained using the cluster operator with a particular "cut" operator) can be re-grouped based on a different "cut" operator, which may depend from the previous cut and/or can be based on other parameters or criteria. For example, these subsets can be merged back into a single set of rules and the cluster operator is then applied to this subset again (but with a different "cut" parameter). Other operators can also be used to regroup the subsets in Set S.

I. Filtering Operators

An exemplary filtering operator receives a subset of rules and splits this subset into at least 2 subsets: one subset contains rules which pass a predetermined selection criteria of the filter, and another subset contains rules which do not. In particular, this selection criteria may be

specified using a data mining query (or a pattern template). The data mining query describes a class of patterns in general terms.

Data mining queries are described in publications - T. Imielinski et al., "DataMine: Application Programming Interface and Query Language for Database Mining", Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, August 1996; J. Han et al., "DMQL: A Data Mining Query Language for relational Databases", Proceedings of the SIGMOD Workshop in Research Issues on Data Mining and Knowledge Discovery, Montreal, June 1996; and W. Shen et al., "Metaqueries for Data Mining," Advances in Knowledge Discovery and Data Mining, chap. 15, AAAI Press, 1996. Any pattern description language or any data mining query language can be used to specify patterns and data mining queries. For example, article by T. Imielinski et al., "DataMine: Application Programming Interface and Query Language for Database Mining," Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, August 1996 introduced "M-SQL" for association rule discovery which is based on software query language ("SQL") modified with additional data mining operators. However, the exemplary embodiment of the data mining query does not depend on any specific language.

For the following exemplary request, "Find all rules in customer purchase data specifying which product categories the customers with children of various ages are buying", M-SQL query is as follows:

```
SELECT *
FROM      Mine(CustomerPurchaseData) R
WHERE     R.Body<{(Children=*), (ChildrenAgeLess6=*),
          (Children6to12=*), (ChildrenAgeMore12=*)} and
          {(Children=*)}<R.Body and R.Consequent IN
          {(CategorySweets=*), (CategoryCereal =*),
          (CategoryFruit=*)} and R.Confidence>=0.5 and
          R.Support>=0.01.
```

This data mining query discovers association rules if and only if they satisfy certain criteria. First, the association rules must include the fields *Children*, *ChildrenAgeLess6*, *ChildrenAgre6to12*, *ChildrenAgeMore12* of the table *CustomerPurchaseData* in the body of the rule. Second, the attribute *Children* must necessarily be present (this is specified by *R. Body*). Third, the discovered patterns must have one of the fields *CategorySweets*, *CategoryCereal* or *CategoryFruit* as a consequent of the rule (specified by *R.Consequent*). Finally, the discovered patterns must satisfy certain thresholds measuring statistical significance (i.e., *R.Confidence* and *R.Support*).

Thus, this exemplary data mining query specifies a set of patterns. The set of these exemplary patterns may indicate:

- the extent to which families with children younger than six years old buy sweets,
- the extent to which families with children older than 12 years old buy sweets,
- the extent to which families with children older than 12 years old buy fruit, etc..

Therefore, the pattern specified by the association rule:

Children = YES **and** *ChildrenAgeLess6* = YES --> *CategorySweets* = *YES (0.01, 0.55)*

noted above is also one of the patterns specified by the data mining query.

Pattern Templates are described in M. Klemmettinen et al., "Finding Interesting Rules for Large Sets of Discovered Association Rules", Proceedings of the Third International Conference on Information and Knowledge Management, December, 1994. For example, a pattern template may be provided as follows:

Children **and** *ChildrenAge* * --> *Category*(0.01,0.5)

where *ChildrenAge* and *Category* are generalizations of attributes. Thus, if *ChildrenAge* specifies the set of attributes {*ChildrenAgeLess6*, *ChildrenAge6to12*, *ChildrenAgeMore12*} and *Category* specifies the set of

attributes {*CategorySweets, CategoryCereal, CategoryFruit*}, then this pattern template specifies the same patterns as the above-described data mining query.

II. Clustering Operator

The clustering operator receives, as input, a subset of rules and an attribute hierarchy of this subset. In particular, the attribute hierarchy can be formed using the procedure described below with reference with Fig. 10. An exemplary attribute hierarchy is illustrated in Fig. 13. All of the fields (i.e., attributes) of the attribute hierarchy are provided at the bottom of the attribute hierarchy. These fields are portions of the transaction file TRANS(Trans_ID, Cust_ID, C_1, ... C_n) as described above, without the fields Trans_ID and Cust_ID. In the exemplary hierarchy illustrated in Fig. 13, n = 13.

A top portion of Fig. 10 shows an exemplary procedure to generate the attribute hierarchy. In step 450, grouping data of a particular subset of rules is determined by combining the fields of the TRANS file (e.g., a table) into groups (e.g., fields C1, C2, C3 illustrated in Fig. 13 are combined into group N1, fields C4 and C5 into group N2, etc.). In step 455, these groups are further combined into larger groups, and so on. For example and as shown in Fig. 13, groups N2 and N3 are combined into group N4, groups N1 and N4 are combined into group N5, fields C9 and C10 are combined into group N6, group N7 and field C11 are combined into group N8, groups N6 and N8 are combined into N9, and groups N5 and N9 are combined into N10. As a result, the attribute hierarchy is generated (step 455), with attributes of the TRANS transaction file being its leaves. It should be noted that a tree which defines this attribute hierarchy (shown in Fig. 13) does not have to be balanced, i.e., all path lengths from the root node to the leaves do not have to be equal.

The attribute hierarchy may include one or more (e.g., two) levels of nodes below the descendent leaves of the attribute hierarchy (i.e., the fields of the TRANS transaction

file). A first level consists of a pair of attributes - field and a relational operator. The relational operator may include exemplary operators such as "=", "<", ">", etc. A second level is below the first level and consists of three attributes - field, relational operator and sets of values which the field attribute can be compared to (e.g., predetermined values, one or more intervals, etc.). For example, the second level can be (C3, =, a) (i.e., field C3 uses the relational operator "=" to be compared to variable "a") , (C5, <, 20) (i.e., field C5, via the relational operator "<" is compared to number 20), (C8, =, [60, 80]) (i.e., field C8, via the relational operator "=" is compared to a range between 60 and 80), etc. Fig. 14 shows an exemplary illustration of the first and second level extensions of node N7. In particular, the first level of field C12 is a leaf 540, which contains field C12 and a relational operator "<". Below leaf 540, a lowest leaf of field C12 (leaf 550) is provided with field C12, the relational operator "<" and a comparison value "20". In addition, the first level of field C13 is a leaf 545, which contains field C13 and a relational operator "=". Below leaf 545, a lowest leaf of field C13 (leaf 555) is provided with field C13, the relational operator "=" and a comparison range "[60, 80]". These leaves are only provided for illustrative purposes, and it should be understood that other combinations of field to relational operators to comparison values/ranges are possible. These hierarchies don't necessarily have to include the same number of extensions/leaves. For example, field C12 may have two extensions, field C4 may have one extension, field C5 can have no extensions and field C6 can have four extensions.

After the attribute hierarchy is generated in step 455 (shown in Fig. 10), "Cut" data is generated with respect to the attribute hierarchy (step 460) by providing a "Cut" in the attribute hierarchy. "Cut" in the attribute hierarchy is defined as a set of nodes of the tree such that a union of all descendant leaves of the nodes which were identified in the

cut consists of all the fields of TRANS transaction file (i.e., $C_1,\ldots,C_n$). An exemplary cut is shown in Fig. 13 which includes the following groups/fields - C1, C2, C3, N4, N6, C11 and N7. In addition, the "Cut" is not limited to the nodes of shown in Fig. 13, and can also include one or two levels below the field levels (shown in Fig. 14). Fig. 11 shows a detailed illustration of step 460 in which "Cut" data is generated. In step 480, the "Cut" is provided to the attribute hierarchy. If the "Cut" is properly specified (e.g., all of the leaves of the attribute hierarchy are above the "Cut", leaves being the lowest level of the attribute hierarchy) in step 485, or if the human expert (or the system) indicates that the "Cut" is unacceptable (step 490), a different "Cut" is created using similar techniques as described above for providing the original cut (step 497) and the procedure is restarted at step 485 with this newly created "Cut". Otherwise, "Cut" data is generated as a function of the "Cut" (step 495) and can be stored in memory for a possible future use.

After the "Cut" data is generated (step 460 in Fig. 10), subsets of the user rules are grouped using "Cut" data and the hierarchy data (step 465), and these grouped subsets are placed into Set S (step 470) to be provided to the human expert.

Thus, the clustering operator consists of steps 460-470. As indicated above, the following data is provided as input to the clustering operator: a) initial set of user rules, b) an attribute hierarchy as described above, and c) the "Cut". The output of the clustering operator is Set S which includes subsets of rules. These subsets are mutually exclusive and collectively exhaustive (e.g., a union of the subsets is equal to all of the rules in Set S).

Fig. 12 shows an exemplary procedure for grouping subsets of the user rules using the "Cut" data as described for step 465 above (Fig. 10). In particular, all user rules are combined from a number of subsets of Set S to form Set A. In step 505, another set (i.e., a Cluster Working Set B) is initialized (e.g., to be an empty set or a null set). A new

rule is then retrieved from Set A (step 510). In step 515, if there are no more rules in Set A to be analyzed or regrouped (e.g., Set A has no more rules or is a null set), the exemplary procedure shown in Fig. 12 is completed. Otherwise, in step 520, it is determined if the new rule corresponds to a class of any existing cluster subset in Set B. If that is the case, the new rule is moved into a "matched" subset in Set B (step 525) and the procedure is directed to step 510. Otherwise, a new cluster subset is created in Set B (step 530), the new rule is moved to the new cluster subset in Set B (step 535), and then the procedure is directed to step 510.

Using the "Cut", two rules are provided to the same class if and only if they have the same structure with respect to the "Cut". In particular, the rules should have the same number of attributes and these attributes, e.g., can be grouped in pairs so that two attributes in the same pair have the same ancestor in the "Cut". For example, the rules:

$$C1 = 5 \text{ and } C4 < 6 \text{ and } C9 > 8 \Rightarrow C12 = 8$$

and

$$C1 > 3 \text{ and } C6 = 5 \text{ and } C10 < 2 \Rightarrow C13 < 7$$

are equivalent because fields C4 and C6 (shown in Fig. 13) have group N4 as an ancestor in the "Cut", rules C9 and C10 have group N6 as an ancestor in the "Cut", and rules C12 and C13 have group N7 as an ancestor in the "Cut". It should be noted that the user rules in the same cluster are "equivalent". As such, a new rule retrieved from Set A can be compared with any rule (or a specific rule) in the related cluster subset in Set B in step 520 shown in Fig. 12. In addition and as shown in Fig. 13, the rules:

$$C2 = 4 \text{ and } C9 = 5 \Rightarrow C12 = 8$$

and

$$C2 = 8 \text{ and } C11 = 3 \Rightarrow C12 = 6$$

are not equivalent because fields C9 and C11 do not have a common ancestor in the "Cut". Accordingly, using the procedure shown in Figs. 10 and 12, the subsets of rules of the generated clusters are provided into the set of regrouped rules generated in step 445 of Fig. 9.

After Set S is split into a subset of clusters, one or more statistics may be generated for each cluster. These statistics may be, e.g.,

- the number of rules per cluster.
- if a component of the rule is an attribute, the ranges of values that such attribute can assume. For example, if the attribute is "Age = a", then it may be preferable to collect statistics on the maximum and minimal values for the age in the rules for that cluster, in addition to the average value and standard deviation for that age.
- for different nodes/groups, how many rules correspond to different attributes for each node/group. For example, for group N6 shown in Fig. 13, it is possible to maintain the number of rules with attribute C9 and the number of rules with attribute C10.
- centers of clusters (calculated, e.g., with the method described above and illustrated in Figs. 4 and 5). These centers can be reported to the human expert.

These exemplary statistics may be utilized by the human expert in step 410 (shown in Fig. 9) to determine which sets of rules the human expert may select for a manual examination.
This completes the description of Fig. 9 and the way rules are examined by the human expert.

If the human expert determines that the clustering of rules based on a particular "Cut" is unsatisfactory, the rules may be regrouped in step 445 using a different "Cut". For example, this different cut would be a finer cut which generated a larger number of clusters (which are smaller in size). This can be done by merging back the clusters of rules obtained with the previous "Cut" (in step 445), returning to step 410 where the human expert marks all the merged rules as

"undecided", and then, in step 445 again, re-cluster rules based on the different (e.g., finer) "Cut".

The process and system according to the present invention can be implemented using, e.g., a graphical user interface ("GUI") which enables the human expert to communicate with the validation system according to the present invention. Using this GUI, the human expert selects a number of operators from a graphical menu of the GUI. Exemplary operators provided on this graphical menu may include a "Filtering" operator, a "Clustering" operator and a "Browsing" operator (e.g., allows the human expert to examine sets of rules generated by the "Clustering" operator or another operator). Other operators can also be included in the graphical menu of the GUI.

Fig. 15 shows an exemplary flow of the process and system according to this embodiment of the present invention. In particular, the user (e.g., human expert) can select the "Filtering" operator from the graphical menu and apply this operator to Set S (step 600). As a part of the filtering operator, the human expert may specify a data mining query which selects "Good", "Bad" or "undecided" rules from Set S. Then, in step 605, "Good" rules are moved from Set S to "Good_Rules" set, and "Bad" rules are moved from Set S to "Bad_Rules" set which is, preferably, automatically saved by the system (e.g., the processor) into a memory device. In step 615, the system may mark the user rules which were determined by the user (or automatically by the system) as "undecided". In step 620, the human expert may apply the remaining "undecided" rules through another filter to again obtain "Good", "Bad" and "undecided" rules (which can be determined using another user-specified data mining query) from the rest of the rules. After the second "Filtering" operator is applied, the system may move "Good" rules from Set S to "Good_Rules" set, and "Bad" rules from Set S to "Bad_Rules" set (step 625). In step 640, the human expert may decide to cluster the remaining "undecided" rules in Set S using the "Clustering" operator (which the human expert

selects from the graphical menu). The "Clustering" operator generates many sets of rules that the user may decide to examine using a graphical browser by selecting a "Browsing" operator from the graphic menu (step 645). The "Browsing" operator allows the user (e.g., the human expert) to examine the clusters of generated user rules by analyzing the statistics (described above) for these clusters. This process of selecting operators (from the graphical menu of available operators) can continue until, e.g., all the rules in Set S have been validated or until the human expert decides to stop the processing of the validation procedure based on at least one of the above-described stopping criteria.

The human expert may apply a number of (e.g., four) operations in sequence (e.g., two filtering operators, one clustering operator, and one browsing operator). This process can also be performed in parallel (e.g., the human expert may decide to perform two filtering operations in parallel and then combine their results).

In another embodiment of the present invention, while the human expert proceeds deeper into an validation process (i.e., performs more iterations of steps 410-430 and 445 shown in Fig. 9), the process steps may be recorded using the GUI interface.

What Is Claimed Is:

1. A method for generating a user profile for a user based on a static profile and a dynamic profile of the user, the static profile including factual user information, the dynamic profile including user dynamic rules as a function of transactional user information, the method comprising the steps of:

a) retrieving the factual user information and the user dynamic rules;

b) generating the static profile as a function of the factual and transactional user information;

c) compressing the user dynamic rules into user aggregated rules;

d) providing the user aggregated rules to the user;

e) user selecting at least one aggregated rule from the user aggregated rules based on a user-desired criteria;

f) matching the user dynamic rules to the at least one selected aggregated rule to generate the dynamic profile; and

g) combining the static profiles and the dynamic profile to form the user profile.

2. The method according to claim 1, wherein step (e) includes the substep of:

h) validating the user aggregated rules in the dynamic profile.

3. The method according to claim 1, wherein step (c) includes the following substeps:

I. determining a plurality of similar dynamic rules from the user dynamic rules,

ii. combining the plurality of similar dynamic rules into at least one corresponding cluster, and

iii. generating the user aggregated rules as a function of the at least one corresponding cluster.

4. The method according to claim 3, wherein the at least one cluster includes a plurality of clusters, and wherein substep (iii) includes the following substeps:

A) determining a first representative rule for each cluster of the plurality of clusters, and

B) if a number of the plurality of clusters is greater that a predetermined threshold number, compressing the plurality of clusters into a smaller number of the plurality of clusters.

5. The method according to claim 4, wherein substep (iii) further includes the following substeps:

C) identifying users providing the first representative rule which corresponds to a particular cluster of the plurality of clusters to form a user cluster,

D) determining a second representative expression for the user cluster,

E) augmenting the first representative rules and the second representative rules to form combined rules, and

F) converting the combined rules into the user aggregated rules.

6. The method according to claim 3, wherein each of the user aggregated rules is determined by obtaining a center of the at least one corresponding cluster.

7. The method according to claim 5, wherein the user aggregated rules include fuzzy logic characteristics.

8. The method according to claim 4, wherein substep (B) includes the following substeps:

I. selecting a first cluster and a second cluster from the plurality of clusters,

II. determining a cluster distance between the first cluster and the second cluster,

III. determining a first size of the first cluster and a second size of the second cluster,

IV. if the cluster distance between the first cluster and the second cluster is smaller than or equal to a predetermined relation between the first size and the second size, merging the first cluster and the second cluster to form a merged cluster, and

V. if the cluster distance is larger than the predetermined relation, selecting a further first cluster and a further second cluster and repeating substeps II through IV using the further first cluster as the first cluster and using the further second cluster as the second cluster.

9. The method according to claim 8, wherein substep (B) includes the following substep:

VI. determining a center cluster of the plurality of clusters from the merged cluster.

10. The method according to claim 1, wherein step (a) further includes a step of retrieving a previous dynamic profile of the user, and wherein step (g) includes a step of combining the previous dynamic profile to the static profile and the dynamic profile to form the user profile.

11. A method for providing suggestions to a user based on a user profile associated with the user, comprising the steps of:

a) receiving user current state information associated with the user;

b) retrieving a static profile associated with the user, the static profile including factual user information corresponding to user preferences;

c) retrieving a dynamic profile associated with the user, the dynamic profile including rules corresponding to user repetitive transactions;

d) compressing the rules of the dynamic profile to form aggregated rules as a function of a predetermined similarity criteria;

e) providing the aggregated rules corresponding to the dynamic profile to an expert;

f) selecting at least one rule from the aggregated rules based on a user-desired criteria;

g) matching user dynamic rules to the at least one selected rule to update the dynamic profile;

h) combining the static profile and the dynamic profile to form the user profile; and

i) providing recommendations to the user as a function of the user profile and the user current state information.

12. The method according to claim 11, further comprising the step of:

j) after step (h) and before step (i), receiving present state-of-the-world information corresponding to the user profile and the user current state information,

wherein step (i) further includes a step of providing recommendations as a further function of the present state-of-the-world information.

13. The method according to claim 12, further comprising the step of:

k) after step (a) and before step (b), receiving past transactional information associated with the user.

14. The method according to claim 12, wherein step (d) includes the following substeps:

i. determining a plurality of similar dynamic rules from the user dynamic rules,

ii. combining the plurality of similar dynamic rules into at least one corresponding cluster, and

iii. generating the user aggregated rules as a function of the at least one corresponding cluster.

15. A system for generating a user profile for a user based on a static profile and a dynamic profile of the user, the static profile including factual user information corresponding to substantially fixed user information, the dynamic profile including user dynamic rules corresponding to transactional user information, the system comprising:

a communication arrangement;

a storage arrangement storing the static and dynamic profiles of the user;

a processor retrieving the factual user information and the user dynamic rules from the storage arrangement, the processor generating the static and dynamic profile as a function of the factual user information, the user dynamic rules being compressed by the processor to form user aggregated rules, the user aggregated rules being provided to the communication arrangement; and

a device coupled to the communication arrangement and including a communications unit, a display unit and an input device, the communications unit receiving the user aggregated rules via the communication arrangement, the display unit displaying the user aggregated rules to at least one of the user and a human expert, the input unit accepting commands from the user to select at least one aggregated rule from at least one of the user and the human expert aggregated rules based on a desired criteria, the communications unit providing the at least one selected aggregated rule to the processor via the communication arrangement,

wherein the processor matches the user dynamic rules to the at least one selected aggregated rule to generate the

dynamic profile, and wherein the static profile and the dynamic profile are combined to form the user profile.

16. The device according to claim 15, wherein the communication arrangement includes a telephone communication line.

17. The device according to claim 15, wherein the communication arrangement includes a wireless communication arrangement.

18. The system according to claim 15, wherein the device collects transactional data from the user and provides the transactional data to the processor via the communication arrangement.

19. A method for providing suggestions to a user based on a user profile associated with the user, comprising the steps of:

a) receiving user current state information and transactional information associated with the user;

b) storing the transactional information as at least a portion of a user purchasing history;

c) providing the user purchasing history to a user profile generation module for generating the user profile as a function of the user purchasing history;

d) estimating user needs information as a function of at least one of the user purchasing history, user current state information and the user profile;

e) generating purchasing recommendations to the user as a function of the user estimated needs information and state-of-the-world information; and

f) providing the recommendations to the user using a remote unit.

20. The method according to claim 19, wherein the recommendations are generated as a further function of the

state-of-the-world information, the state-of-the-world information including at least one of product-service location information, discount information and price information.

21. The method according to claim 19, wherein the user profile includes a static profile and a dynamic profile.

22. The method according to claim 19, wherein the user profile includes user rules, the user rules being matched to the user purchasing history and the state-of-the-world information for generating the user needs information.

23. The method according to claim 21, wherein the dynamic profile is improved by compressing user dynamic rules to form user aggregated rules.

24. The method according to claim 19, further comprising the steps of:

(g) before step (b) and after step (a), transmitting the transactional information via a telecommunications arrangement from the remote device to a central computing device; and

(h) before step (f) and after step (e), transmitting the recommendations via the telecommunications arrangement from the central computing device to the remote unit.

25. The method according to claim 19, further comprising the steps of:

(I) before step (e) and after step (d), transmitting the user estimated needs information via a telecommunications arrangement from the remote device to a central computing device; and

(j) before step (f) and after step (e), transmitting the recommendations via the telecommunications arrangement from the central computing device to the remote unit.

26. A system for providing suggestions to a user based on a user profile associated with the user, comprising:

a first module receiving user current state information and transactional information associated with the user;

a storage device storing as at least a portion of a user purchasing history obtained from the transactional information;

a second module receiving the user purchasing history and generating the user profile as a function of the user purchasing history;

a third module estimating the user needs information as a function of the user purchasing history, the user profile and the user current state information;

a fourth module generating recommendations to the user as a function of the user estimated needs information and state-of-the-world information; and

an output device providing the recommendations to the user using the remote unit.

27. The system according to claim 26, wherein the system is a personal shopping assistant system providing recommendations to the user.

28. The system according to claim 27, wherein the remote unit includes a personal digital assistant device accepting commands from the user and providing a currents state of the user to the personal shopping assistant system.

29. The system according to claim 26, wherein the user profile includes a static profile and a dynamic profile, the dynamic profile being improved by compressing user dynamic rules to form user aggregated rules.

30. The method according to claim 26,

wherein the remote unit includes the output device and a fifth module transmitting the transactional information via a telecommunications arrangement from the remote device to a central computing device, and

wherein the central computing device includes the first, second, third and fourth module and the storage device.

31. The method according to claim 26,

wherein the remote unit includes the first, second and third module, the storage device, the output device, and

wherein the fourth module is provided in a central computing device, the remote unit transmitting the user estimated needs information via a telecommunications arrangement to the central computing device, the central computing device transmitting the recommendations to the remote unit.

32. A method for validating user rules using a processing device, the user rules being indicative of at least one of factual user information and transactional user information, the method comprising the steps of:

a) retrieving the user rules from a storage device;

b) separating the user rules into at least one subset of a user set;

c) determining if particular rules of the at least one subset is one of acceptable, unacceptable and undecided based on a defined criteria; and

d) if the particular rules of at least one subset are acceptable, providing the particular rules of the at least one subset to a corresponding user.

33. The method according to claim 32, further comprising the steps of:

f) if the particular rules are acceptable, moving the at least one subset from the user set to a further set; and

g) if the particular rules are unacceptable, removing the at least one subset from the user set.

34. The method according to claim 32, wherein the factual user information is a part of a static profile, wherein the transactional user information is utilized to provide a

dynamic profile which includes the user rules, and wherein step includes the step of generating a user profile for the corresponding user based on the static profile and the dynamic profile of the corresponding user, the generating step including the substeps of:

i) retrieving the factual user information and the user dynamic rules,

ii) generating the static profile as a function of the factual and transactional user information,

iii) compressing the user dynamic rules into user aggregated rules,

iv) providing the user aggregated rules to the user,

v) selecting, by a user, at least one aggregated rule from the user aggregated rules based on a further criteria,

vi) matching the user dynamic rules to the at least one selected aggregated rule to generate the dynamic profile, and

vii) combining the static profiles and the dynamic profile to form the user profile.

35. The method according to claim 32, wherein the at least one subset includes a plurality of subsets, and wherein steps (c) and (d) are executed for each of the subsets.

36. The method according to claim 35, further comprising the steps of:

i) forming the user set with the subsets using a predetermined criteria.

37. The method according to claim 32, further comprising the step of:

j) displaying the particular rules of the at least one set; and

k) if the particular rules in the at least one subsets are determined to be one of acceptable and unacceptable,

correspondingly marking the at least one set as one of an acceptable subset and an unacceptable subset.

38. The method according to claim 37, wherein the at least one subset is displayed on a display device.

39. The method according to claim 32, further comprising the steps of:

l) if the particular rules of the at least one subset are marked as unacceptable, re-forming the user set using the at least one subset to generate further subsets of the re-formed user set; and

m) repeating steps (c) and (d) using each of the further subsets as the at least one subset.

40. The method according to claim 32, further comprising the steps of:

n) if the particular rules of the at least one subset are marked as undecided, re-forming the user set using the at least one subset to generate further subsets of the re-formed user set; and

o) repeating steps (c) and (d) using each of the further subsets as the at least one subset.

41. The method according to claim 40, wherein step (n) includes the substeps of:

i) combining attributes from the transactional user information into related groups,

ii) generating attribute hierarchy data using the related groups,

iii) configuring at least a portion of the attribute hierarchy data as a function of separation data, the separation data separating a first portion of the attributed hierarchy data from a second portion of the attribute hierarchy data, and

iv) grouping the user rules into further subsets of the user set as a function of the attribute hierarchy data and the separation data.

42. The method according to claim 41, wherein the separation data is generated based on the attribute hierarchy data and using a further criteria which is defined by a human expert.

43. The method according to claim 42,
wherein substep (iii) includes further substeps of:

A. generating a cut to separate the attribute hierarchy data, and

B. if the cut is improperly specified or if the cut is unacceptable, generating a new cut and repeating substep A using new cut,

wherein the separation data is generated as a function of at least one of the cut and the different cut.

44. The method according to claim 42, wherein substep (iv) includes the substeps of:

A. combining the user rules from the user set to form a first set,

B. initializing a second set,

C. retrieving a further rule from the first set,

D. if the first set does not have any rules, aborting substep (iv),

E. if the further rule corresponds to a predetermined class of a particular cluster subset in the second set, moving the further rule from the first set to the particular cluster subset, and

F. if the further rule does not correspond to the predetermined class of the particular cluster subset in the second set, generating a new cluster subset in the second set having a further class which

corresponds to the further rule, and moving the further rule from the first set to the new cluster subset.

45. The method according to claim 40, wherein the at least one subset includes a plurality of subsets, and wherein step (n) includes the substep of:

i) selecting a predetermined number of the subsets in the user set, and

ii) merging the predetermined number of subsets to form the further subsets.

46. The method according to claim 45, wherein the predetermined number of the subsets is selected by one of a human expert and as a function of a predetermined selection criterion.

47. The method according to claim 46, wherein the predetermined selection criteria provides that a size of the subsets is smaller than a predetermined value.

48. The method according to claim 40, wherein the at least one subset includes a plurality of subsets, and wherein step (n) includes the substep of:

i) selecting a predetermined number of the subsets in the user set, and

ii) applying a partitioning operator to each of the selected subsets.

49. The method according to claim 48, wherein the partitioning operator includes at least one of a filtering operator and a clustering operator.

50. The method according to claim 49, wherein substep (ii) includes the substeps of:

receiving, at the filtering operator, a particular subset of the subsets, and

separating the particular subset from a first subset and a second subset of the user set, the first subset including the user rules which pass a predetermined selection criteria of the filtering operator, the second subset including the user rules which do not pass the predetermined selection criteria.

51. The method according to claim 50, wherein the predetermined selection criteria is specified using at least one of a data mining query and a pattern template.

52. The method according to claim 32, further comprising the step of:

p) terminating an operation of the method based on a predetermined condition.

53. The method according to claim 52, wherein the predetermined condition includes at least one of:

- a first condition in which the user set is empty,
- a second condition in which a number of subsets in user set is less than a first predetermined value,
- a third condition in which a ratio of the user rules in user set with respect to all of existing rules is less than a second predetermined value, and
- a fourth condition in which a human expert stops the operation of the method.

54. An arrangement for validating user rules, the user rules being indicative of at least one of factual user information and transactional user information, the arrangement comprising:

a storage device maintaining the user rules; and

a processing device retrieving the user rules from a storage device and providing the user rules into at least one subset of a user set,

wherein the processing device determines if particular rules of the at least one subset are one of acceptable, unacceptable and undecided based on a defined criteria,

wherein the processing device providing the particular rules of the at least one subset to a corresponding user if the particular rules of at least one subset are acceptable.

55. The arrangement according to claim 54,

wherein the processing device moves the at least one subset from the user set to a further set if the particular rules are acceptable, and

wherein the processing device removes the at least one subset from the user set if the particular rules are unacceptable.

56. The arrangement according to claim 55,

wherein the factual user information is a part of a static profile,

wherein the transactional user information is utilized to provide a dynamic profile which includes the user rules, and

wherein the processing device generates a user profile for the corresponding user based on the static profile and the dynamic profile of the corresponding user, the processing device providing the particular rules of the at least one subset to the corresponding user by:

i) retrieving the factual user information and the user dynamic rules,

ii) generating the static profile as a function of the factual and transactional user information,

iii) compressing the user dynamic rules into user aggregated rules,

iv) providing the user aggregated rules to the user,

v) selecting, by a user, at least one aggregated rule from the user aggregated rules based on a further criteria,

vi) matching the user dynamic rules to the at least one selected aggregated rule to generate the dynamic profile, and

vii) combining the static profiles and the dynamic profile to form the user profile.

57. The arrangement according to claim 54, wherein the at least one subset includes a plurality of subsets, and wherein the processing device determines and provides the acceptable subset for each of the subsets.

58. The arrangement according to claim 57, wherein the processing device forms the user set with the subsets using a further criteria.

59. The arrangement according to claim 54,

wherein the processing device outputs the particular rules of at least one set, and

wherein the processing device correspondingly marks the at least one set as one of an acceptable subset and an unacceptable subset if the particular rules in the at least one subsets are determined to be one of acceptable and unacceptable.

60. The arrangement according to claim 59, further comprising:

a display device for displaying the at least one subset.

61. The arrangement according to claim 54,

wherein the processing device re-form the user set using the at least one subset to generate further subsets of the re-formed user set if the particular rules of the at least one subset are marked as unacceptable, and

wherein the processing device provides the particular rules of the at least one subset to the corresponding user for each of the further subsets as the at least one subset.

62. The arrangement according to claim 54,

wherein the processing device re-form the user set using the at least one subset to generate further subsets of the re-formed user set if the particular rules of the at least one subset are marked as undecided, and

wherein the processing device provides the particular rules of the at least one subset to the corresponding user for each of the further subsets as the at least one subset.

63. The arrangement according to claim 62, wherein the processing device groups the at least one subset into the further subsets by:

i) combining attributes from the transactional user information into related groups,

ii) generating attribute hierarchy data using the related groups,

iii) configuring at least a portion of the attribute hierarchy data as a function of separation data, the separation data separating a first portion of the attributed hierarchy data from a second portion of the attribute hierarchy data, and

iv) grouping the user rules into further subsets of the user set as a function of the attribute hierarchy data and the separation data.

64. The arrangement according to claim 63, wherein the separation data is generated based on the attribute hierarchy data and using a further criteria which is defined by a human expert.

65. The arrangement according to claim 64,

wherein the processing device configures the portion of the attribute hierarchy data by:

generating a cut to separate the attribute hierarchy data, and

if the cut is improperly specified or if the cut is unacceptable, generating a new cut and separating the attribute hierarchy using new cut,

wherein the separation data is generated as a function of at least one of the cut and the different cut.

66. The arrangement according to claim 64, wherein the processing device groups the user rules into the further subsets by:

combining the user rules from the user set to form a first set,

initializing a second set,

retrieving a further rule from the first set,

if the first set does not have any rules, aborting grouping of the user rules,

if the further rule corresponds to a predetermined class of a particular cluster subset in the second set, moving the particular rules from the first set to the particular cluster subset, and

if the further rule does not correspond to the predetermined class of the particular cluster subset in the second set, generating a new cluster subset in the second set having a further class which corresponds to the further rule, and moving the further rule from the first set to the new cluster subset.

67. The arrangement according to claim 62, wherein the at least one subset includes a plurality of subsets, and wherein the processing device re-forms the at least one subset into the further subsets by:

i) selecting a predetermined number of the subsets in the user set, and

ii) merging the predetermined number of subsets to form the further subsets.

68. The arrangement according to claim 67, wherein the predetermined number of the subsets is selected by one of a human expert and as a function of a predetermined selection criterion.

69. The arrangement according to claim 68, wherein the predetermined selection criteria provides that a size of the subsets is smaller than a predetermined value.

70. The arrangement according to claim 62, wherein the at least one subset includes a plurality of subsets, and wherein the processing device groups the at least one subset into the further subsets by:

i) selecting a predetermined number of the subsets in the user set, and

ii) applying a partitioning operator to each of the selected subsets.

71. The arrangement according to claim 70, wherein the partitioning operator includes at least one of a filtering operator and a clustering operator.

72. The arrangement according to claim 71, wherein the processing device applies the partitioning operator by:

receiving, at the filtering operator, a particular subset of the plurality of subsets, and

separating the particular subset from a first subset and a second subset, the first subset including the user rules which pass a predetermined selection criteria of the filtering operator, the second subset including the user rules which do not pass the predetermined selection criteria.

73. The arrangement according to claim 72, wherein the predetermined selection criteria is specified using at least one of a data mining query and a pattern template.

74. The arrangement according to claim 73, wherein the processing device stops processing the subsets based on a predetermined condition.

75. The arrangement according to claim 74, wherein the predetermined condition includes at least one of:

- a first condition in which the user set is empty,
- a second condition in which a number of subsets in user set is less than a first predetermined value,
- a third condition in which a ratio of the user rules in user set with respect to all of existing rules is less than a second predetermined value, and
- a fourth condition in which a human expert stops the processing device from processing the subsets.

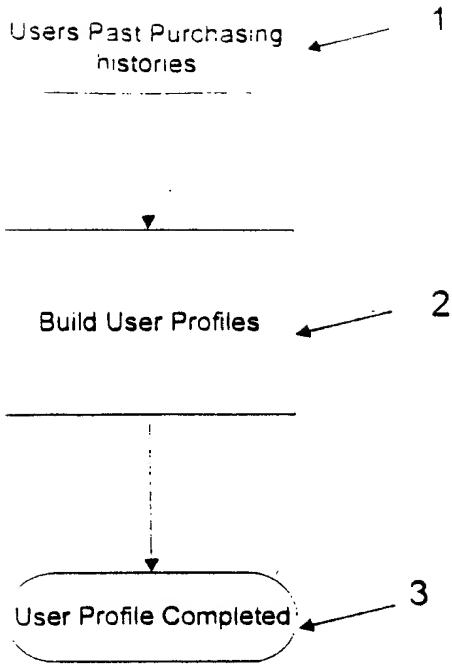


FIGURE 1

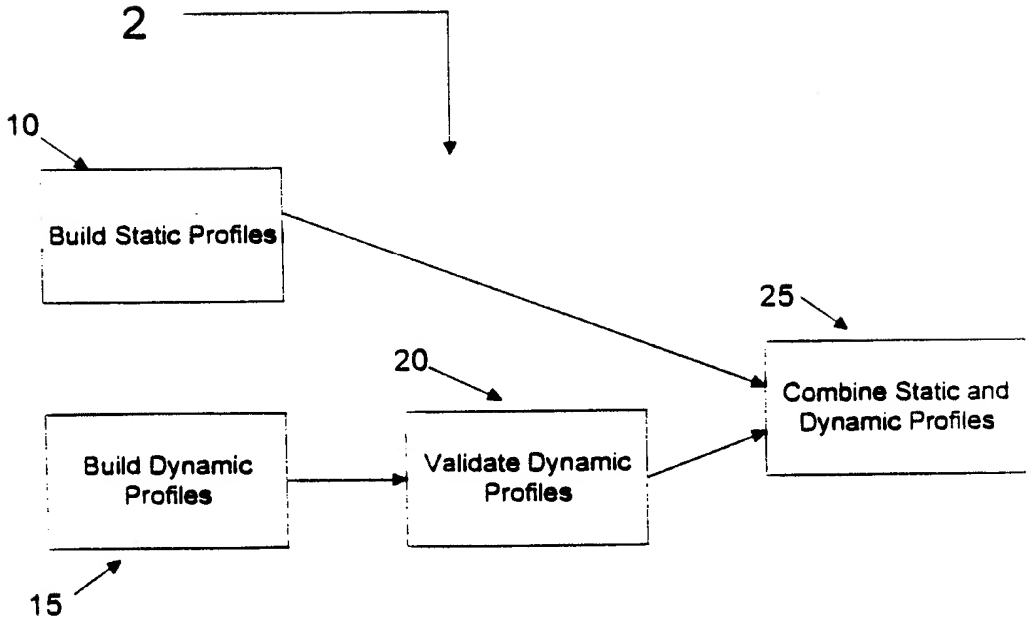


FIGURE 2

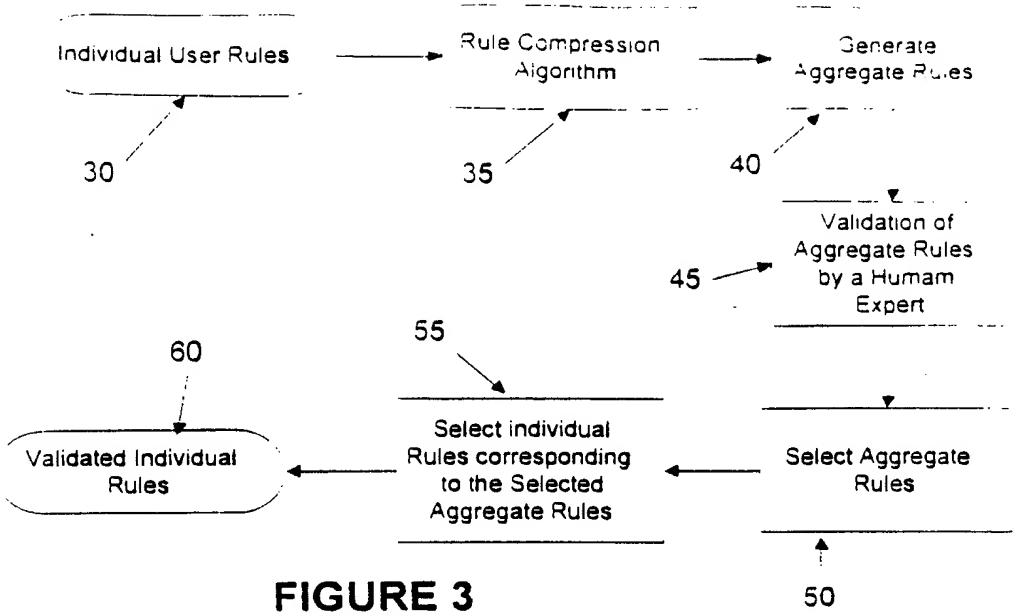


FIGURE 3

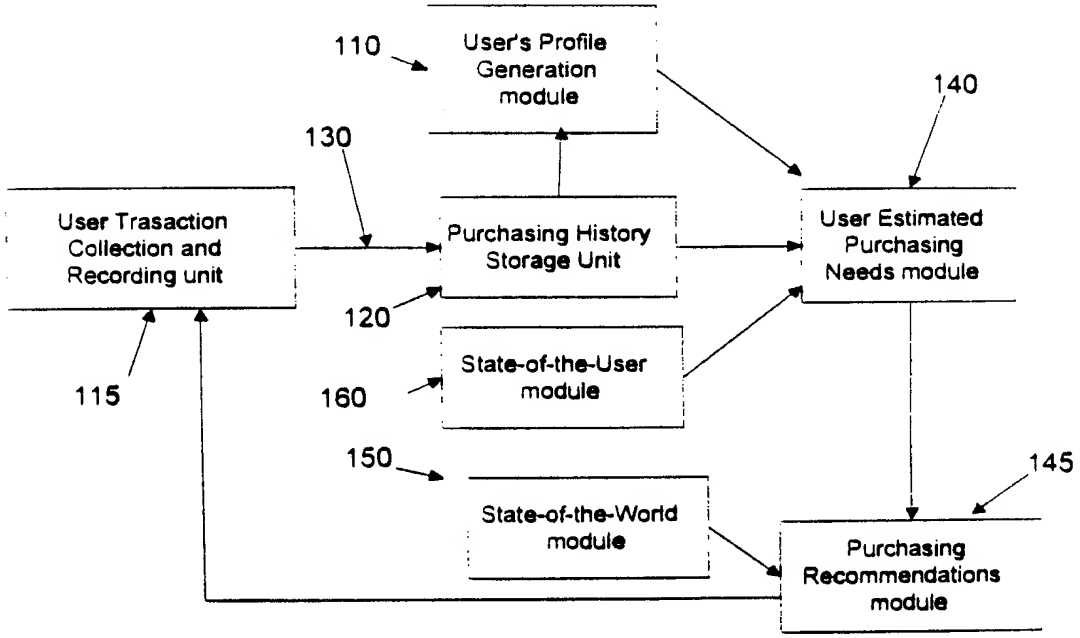


FIGURE 6a

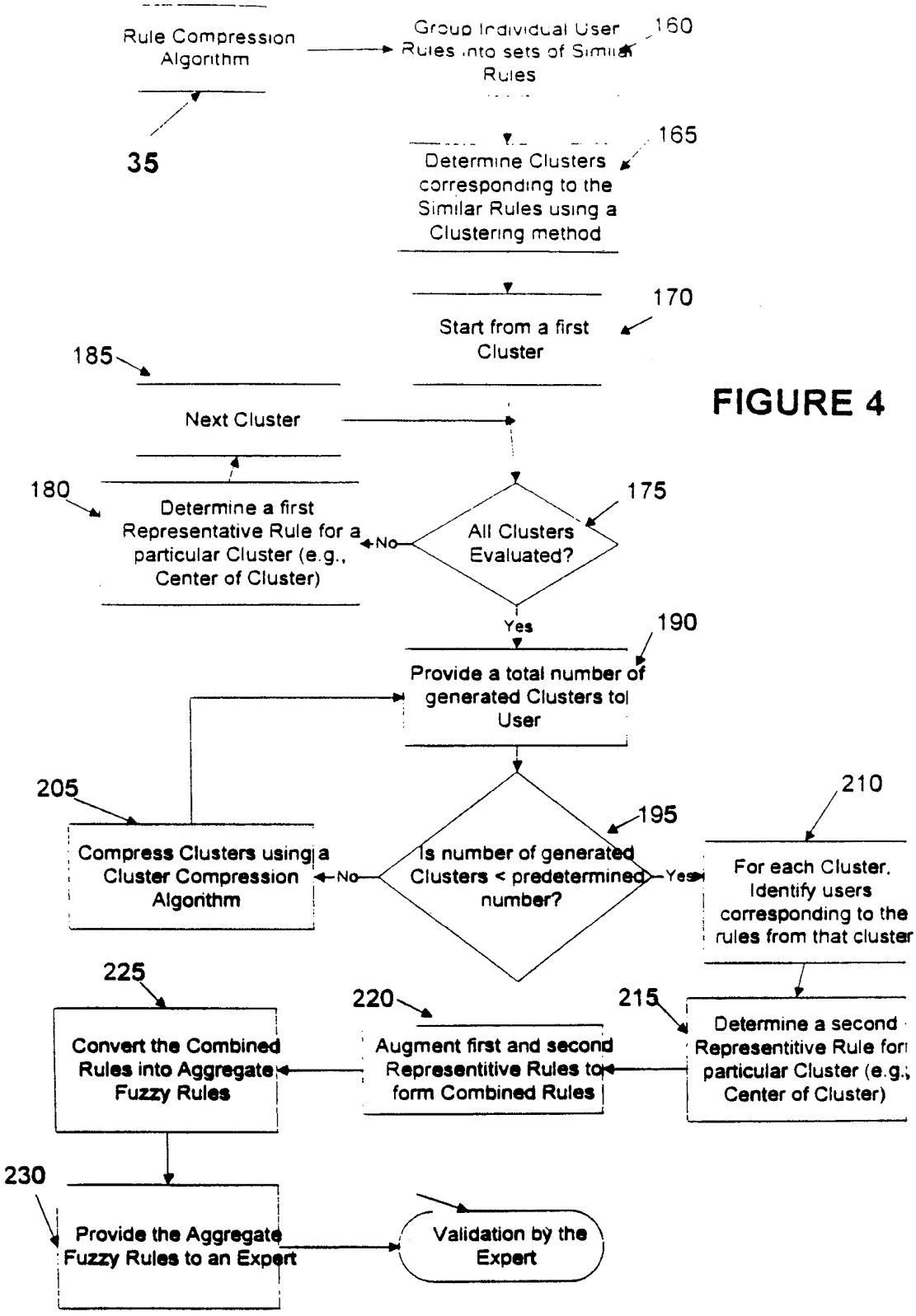

Rule Compression Algorithm
35
Group Individual User Rules into sets of Similar Rules
160
Determine Clusters corresponding to the Similar Rules using a Clustering method
165
Start from a first Cluster
170
Next Cluster
185
Determine a first Representative Rule for a particular Cluster (e.g., Center of Cluster)
180
All Clusters Evaluated?
175
No
Yes
Provide a total number of generated Clusters to User
190
Is number of generated Clusters < predetermined number?
195
No
Yes
Compress Clusters using a Cluster Compression Algorithm
205
For each Cluster, Identify users corresponding to the rules from that cluster
210
Determine a second Representitive Rule for particular Cluster (e.g., Center of Cluster)
215
Augment first and second Representitive Rules to form Combined Rules
220
Convert the Combined Rules into Aggregate Fuzzy Rules
225
Provide the Aggregate Fuzzy Rules to an Expert
230
Validation by the Expert

FIGURE 4

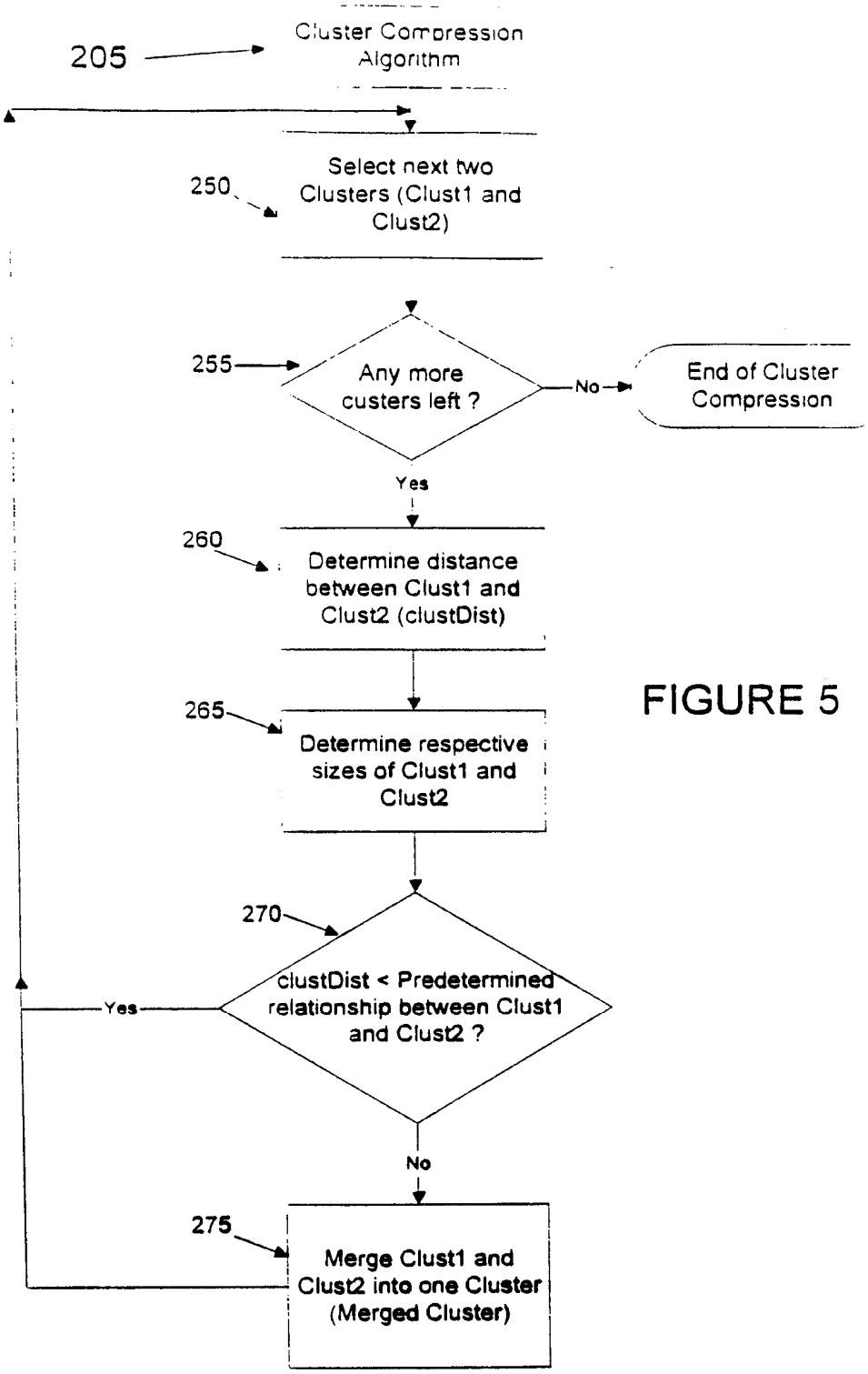

205
Cluster Compression Algorithm
250
Select next two Clusters (Clust1 and Clust2)
255
Any more custers left ?
No
End of Cluster Compression
Yes
260
Determine distance between Clust1 and Clust2 (clustDist)
265
Determine respective sizes of Clust1 and Clust2
270
clustDist < Predetermined relationship between Clust1 and Clust2 ?
Yes
No
275
Merge Clust1 and Clust2 into one Cluster (Merged Cluster)

FIGURE 5

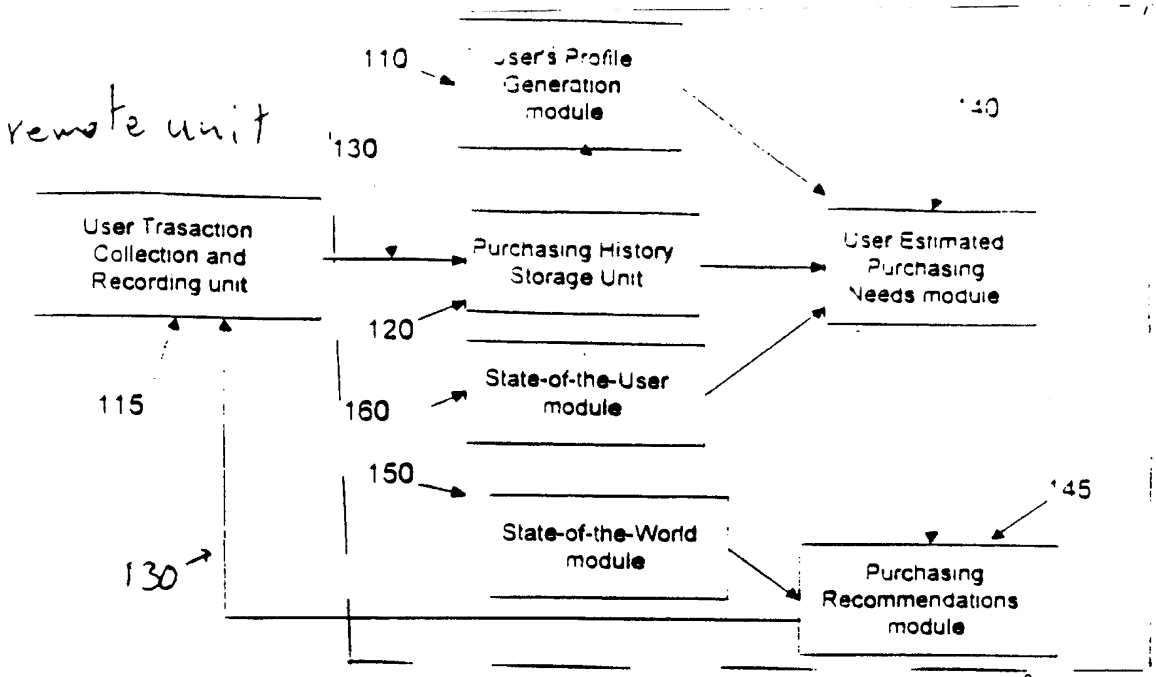


FIGURE 6B

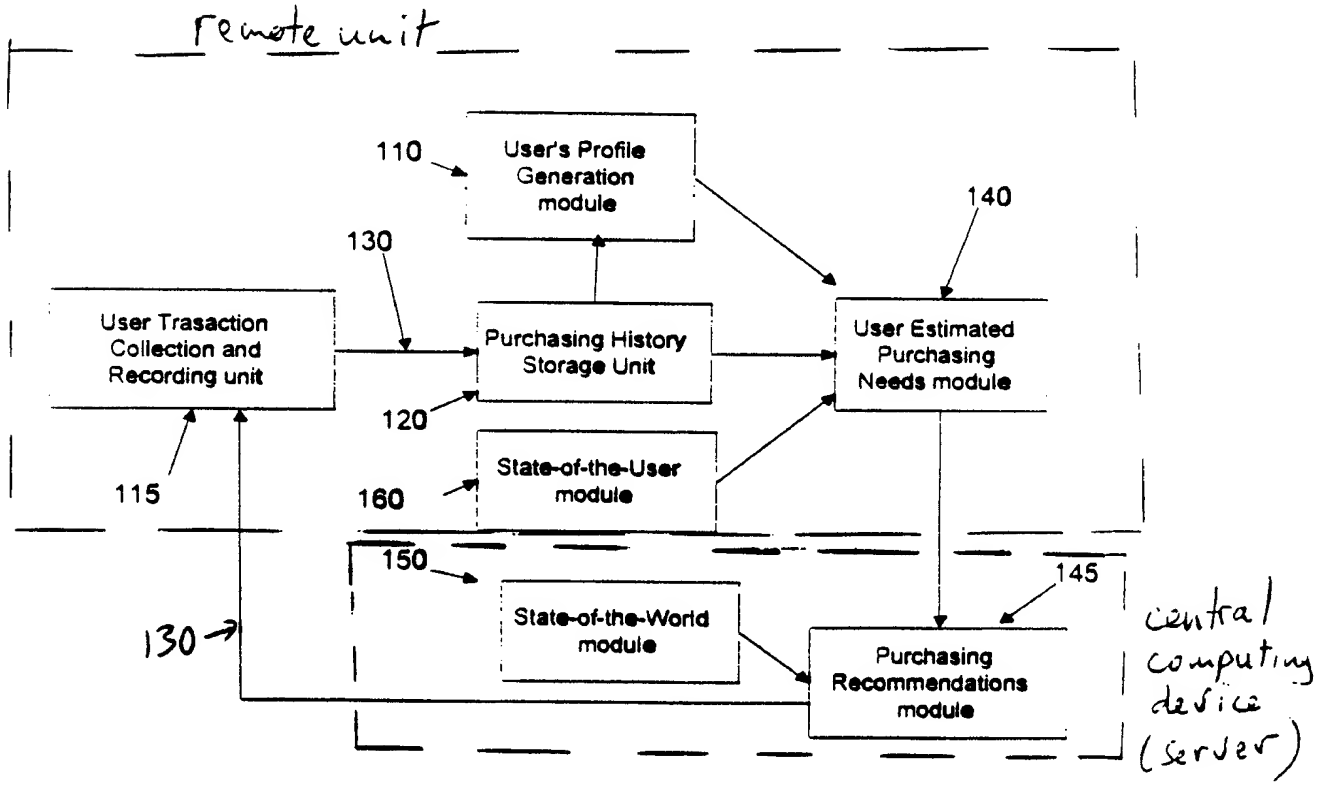


FIGURE 6C

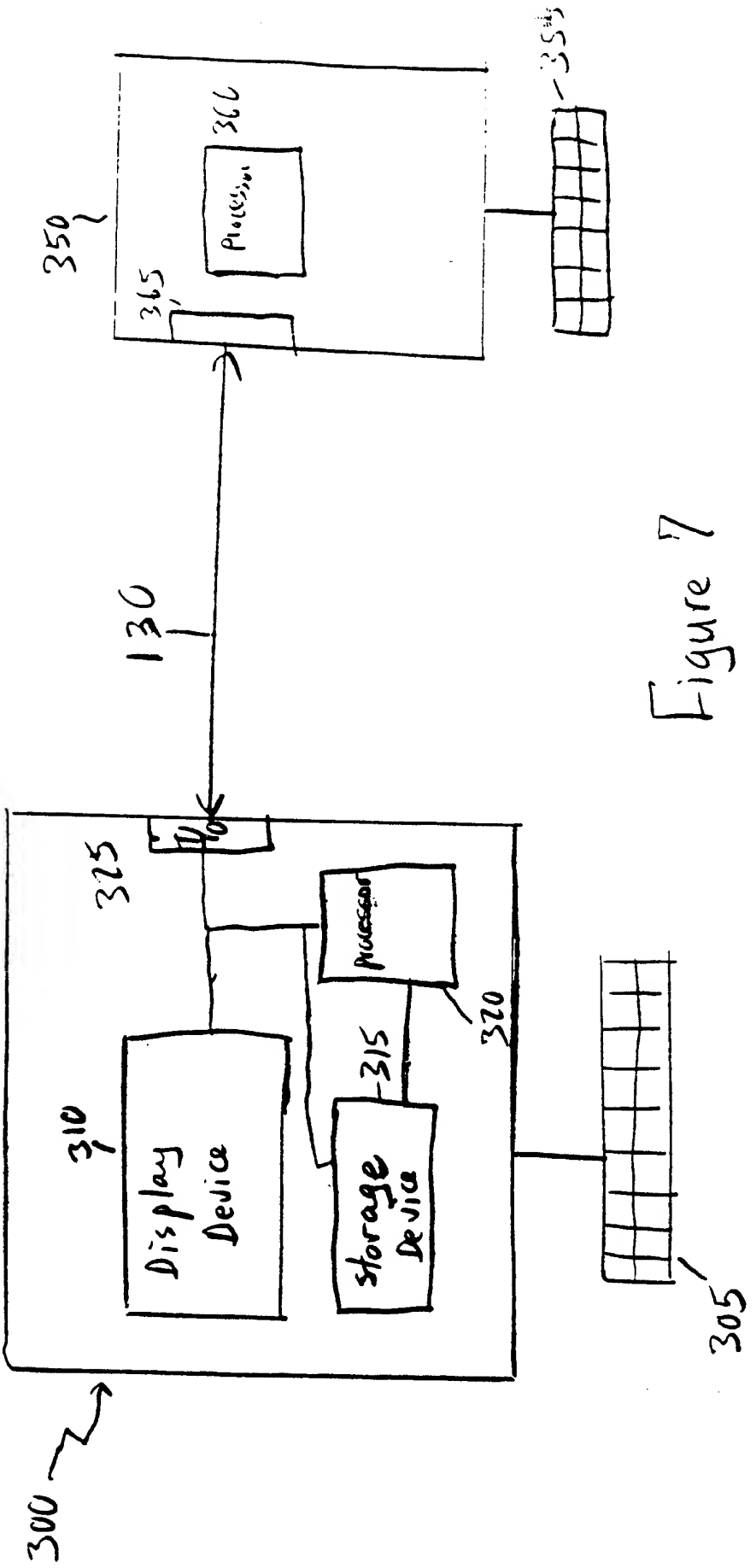

300
310
Display Device
315
Storage Device
320
Processor
325
I/O
130
350
365
366
Processor
355
305

Figure 7

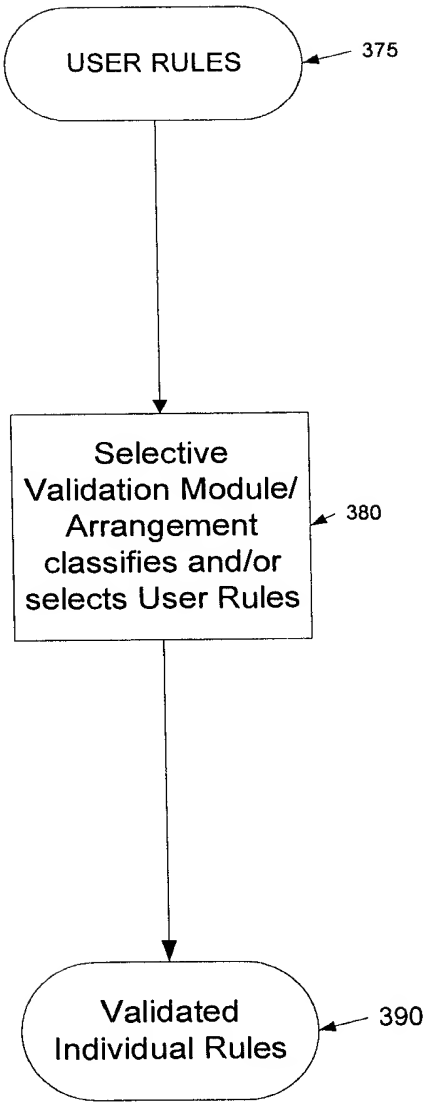

USER RULES
375
Selective Validation Module/ Arrangement classifies and/or selects User Rules
380
Validated Individual Rules
390

FIGURE 8

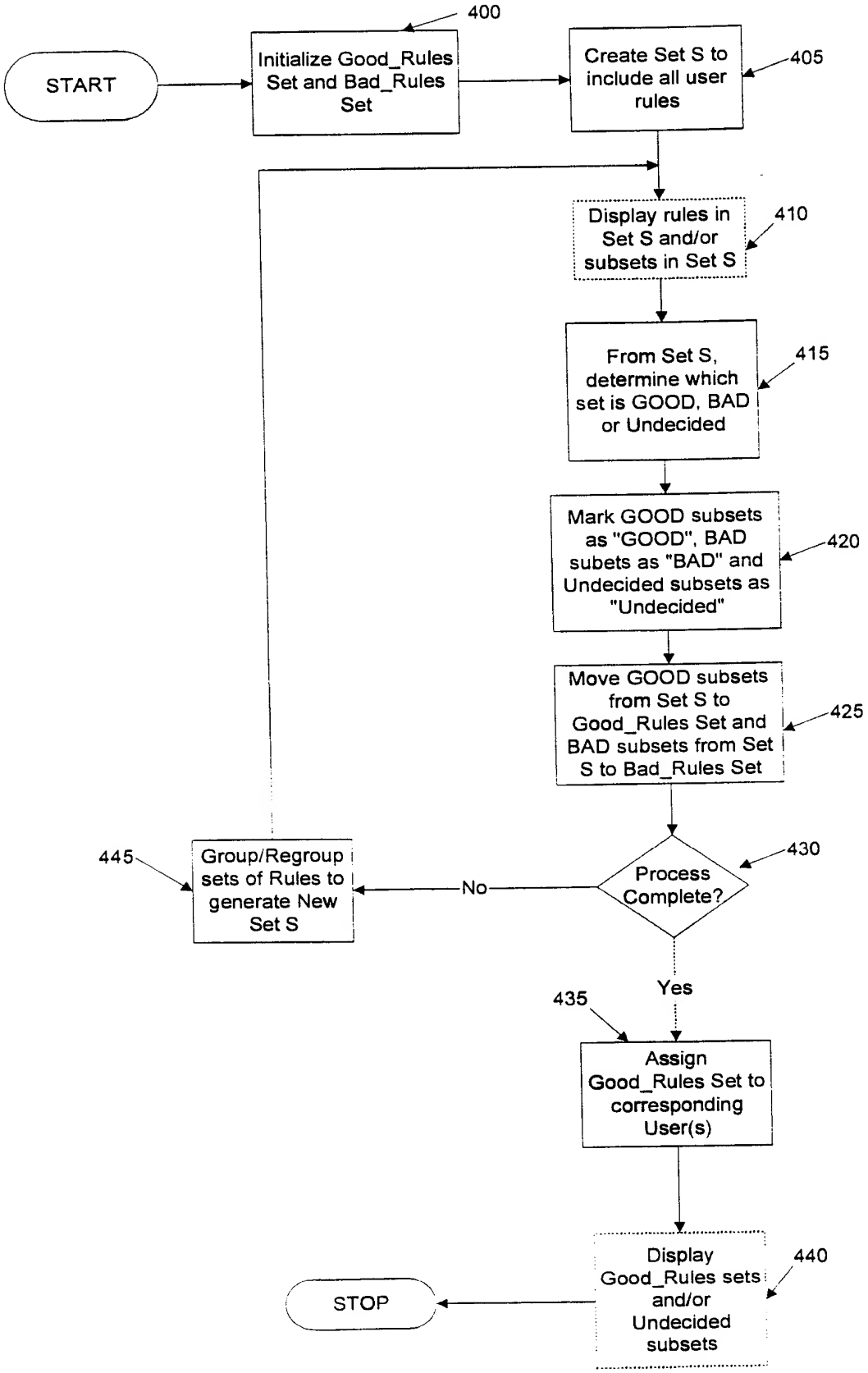

START
400
Initialize Good_Rules Set and Bad_Rules Set
Create Set S to include all user rules
405
Display rules in Set S and/or subsets in Set S
410
From Set S, determine which set is GOOD, BAD or Undecided
415
Mark GOOD subsets as "GOOD", BAD subets as "BAD" and Undecided subsets as "Undecided"
420
Move GOOD subsets from Set S to Good_Rules Set and BAD subsets from Set S to Bad_Rules Set
425
Process Complete?
430
No
Group/Regroup sets of Rules to generate New Set S
445
Yes
435
Assign Good_Rules Set to corresponding User(s)
Display Good_Rules sets and/or Undecided subsets
440
STOP

FIGURE 9

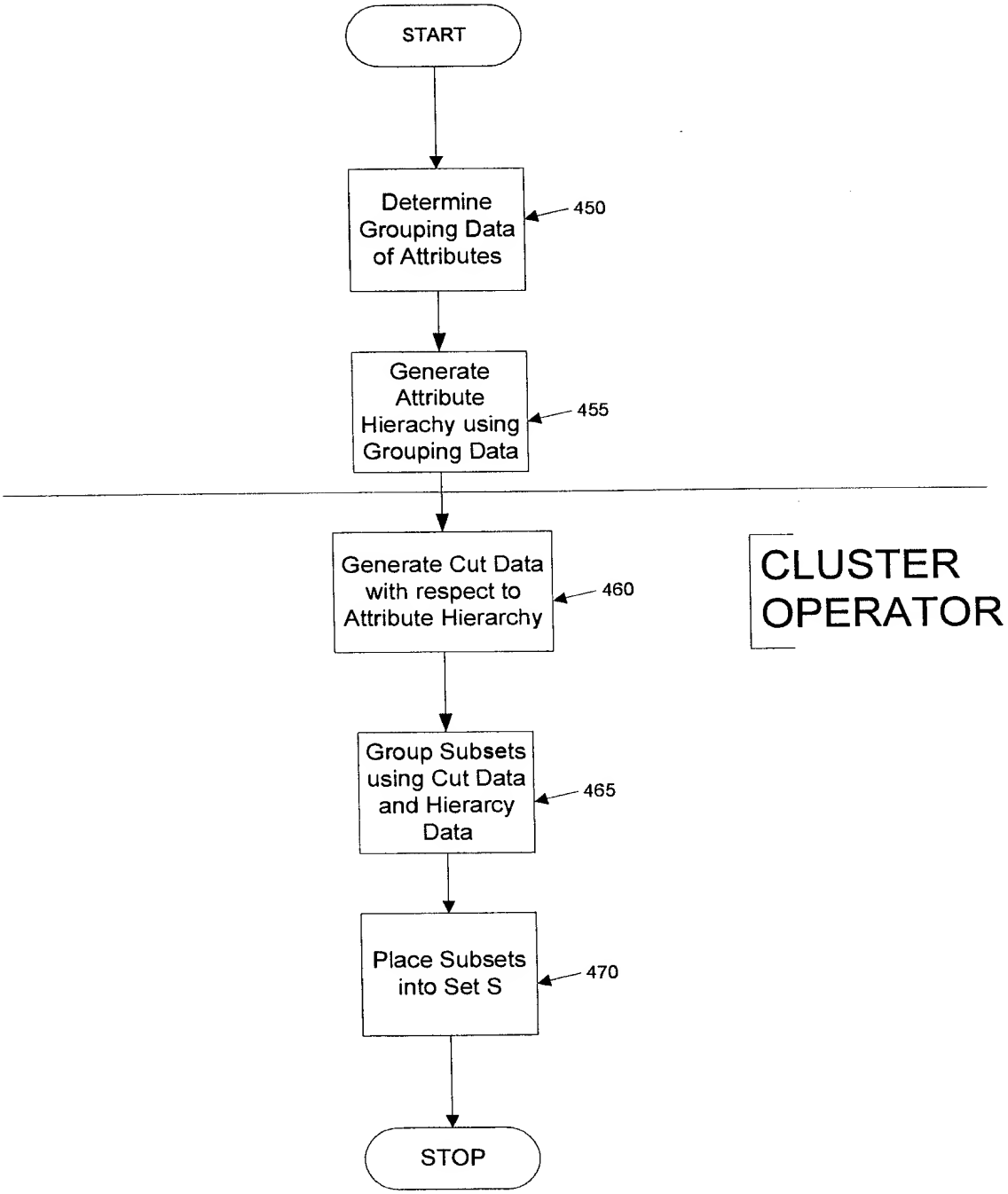

START
Determine Grouping Data of Attributes
450
Generate Attribute Hierachy using Grouping Data
455
Generate Cut Data with respect to Attribute Hierarchy
460
CLUSTER OPERATOR
Group Subsets using Cut Data and Hierarcy Data
465
Place Subsets into Set S
470
STOP

FIGURE 10

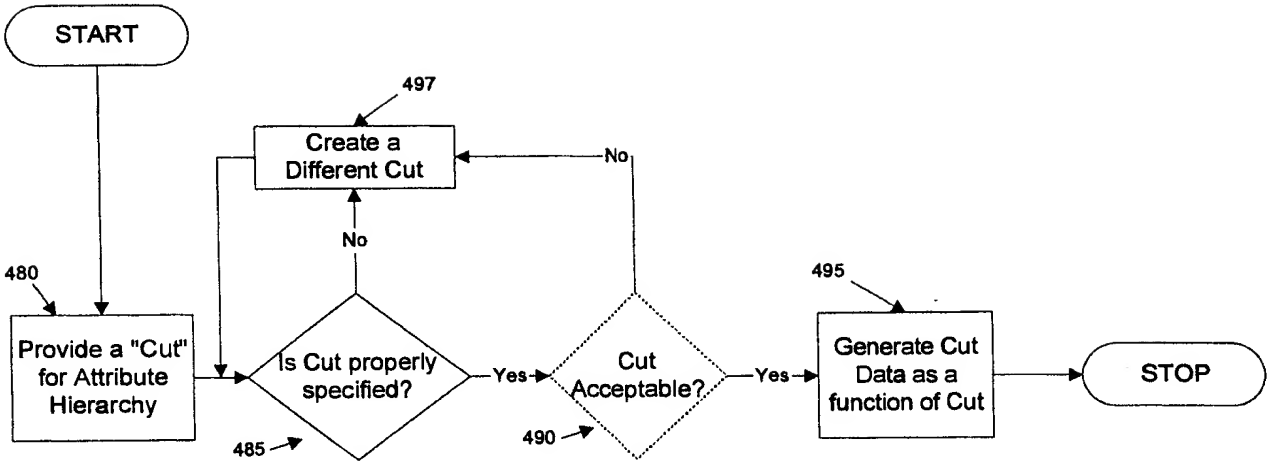


FIGURE 11

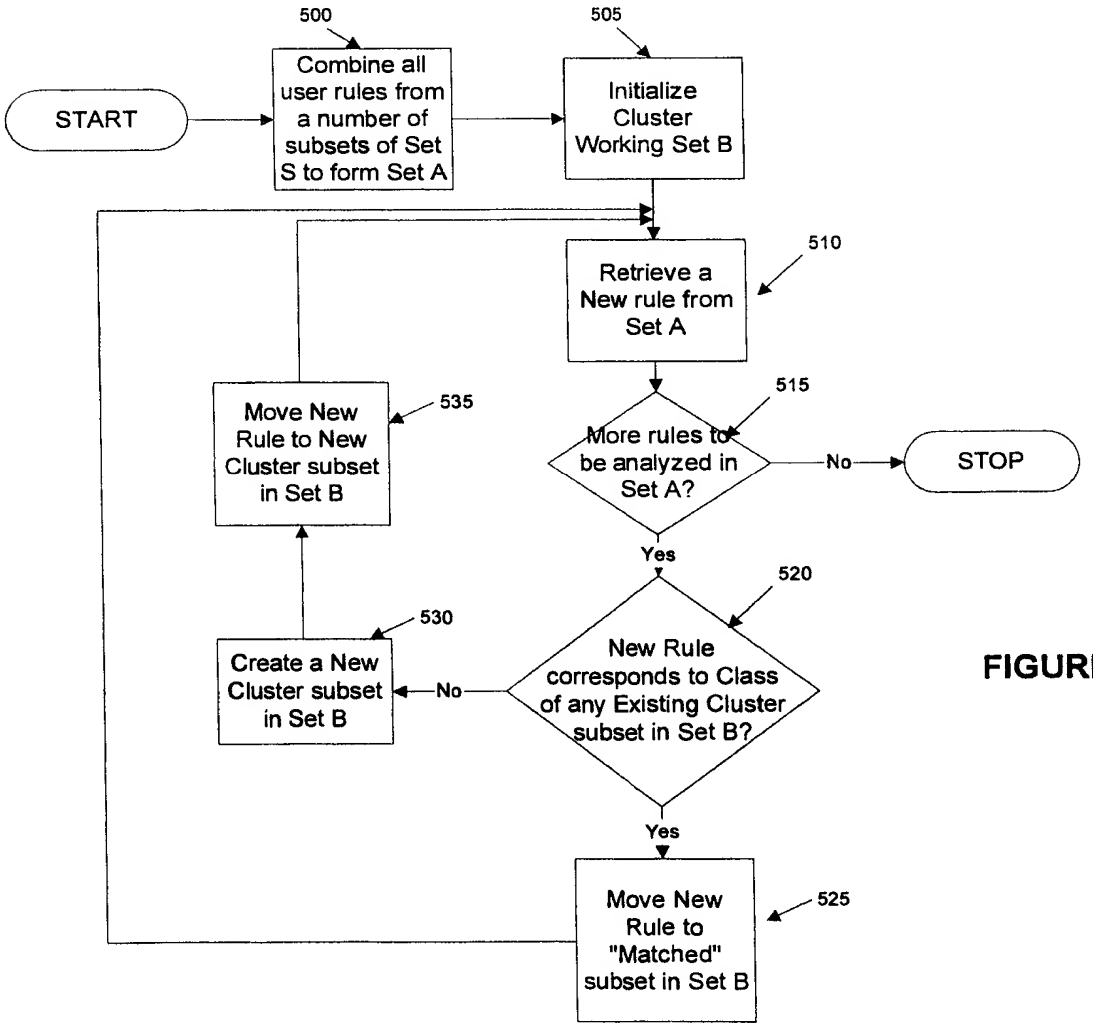


FIGURE 12

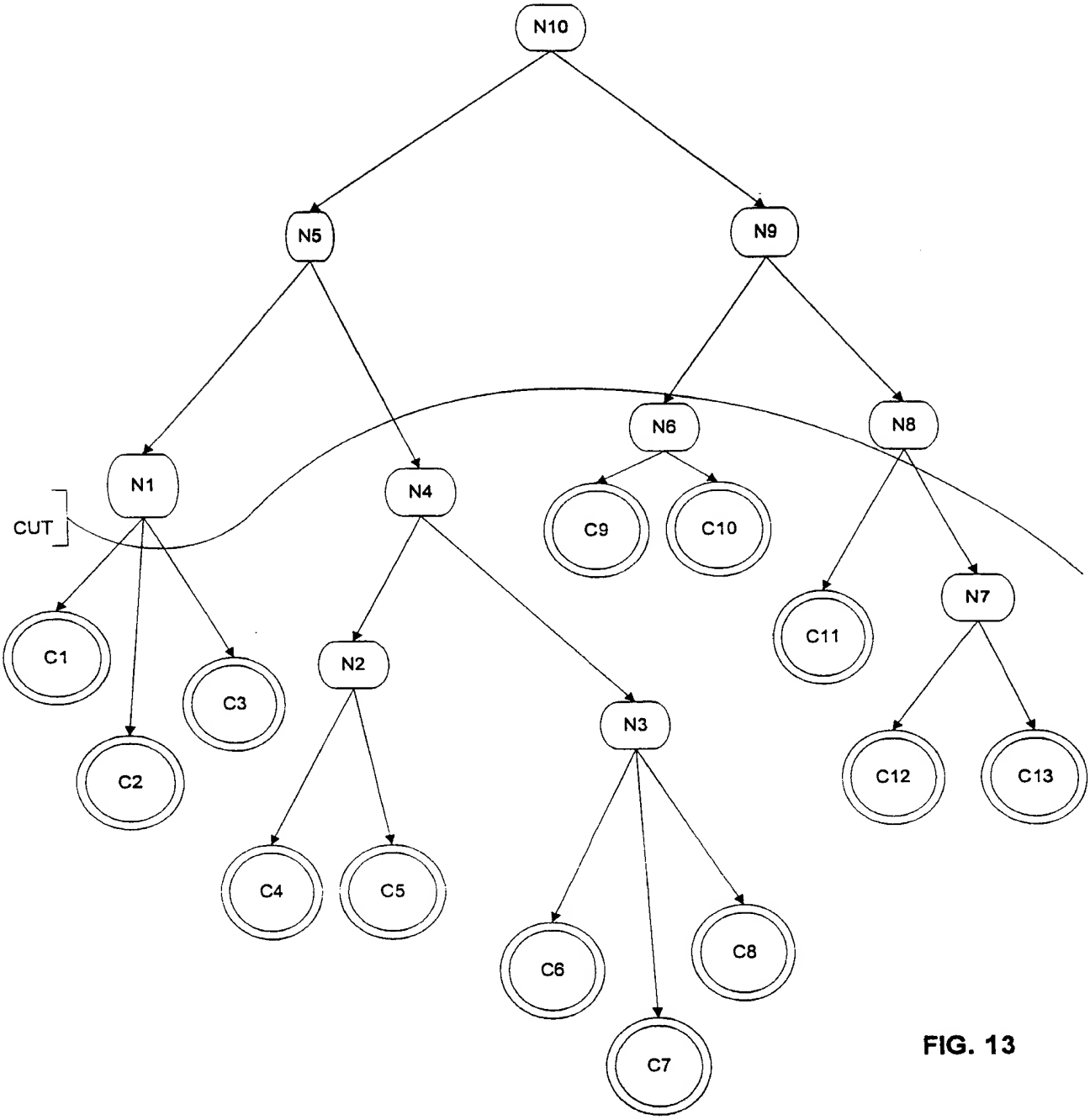

N10
N5
N9
N1
N4
N6
N8
CUT
C1
C2
C3
N2
N3
C4
C5
C6
C7
C8
C9
C10
C11
N7
C12
C13

FIG. 13

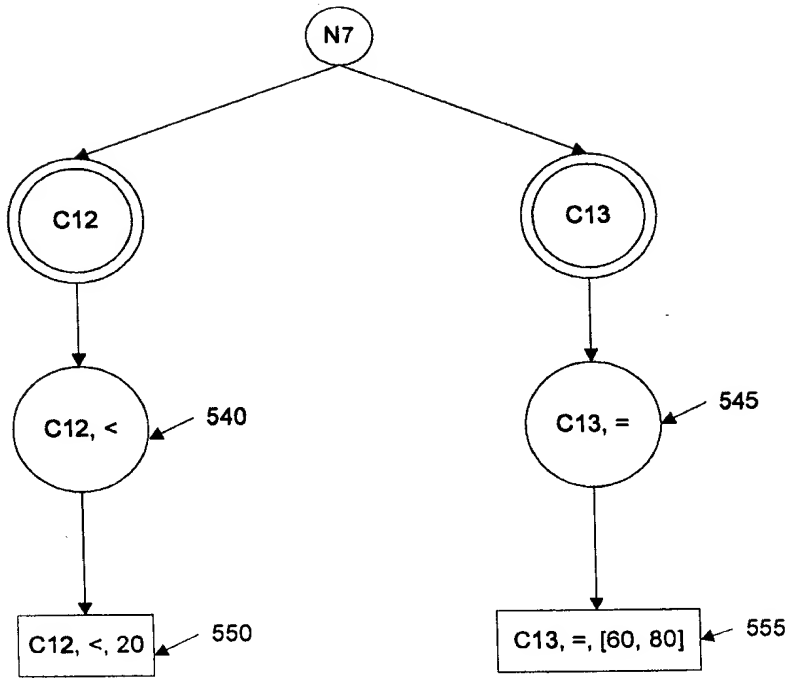


FIGURE 14

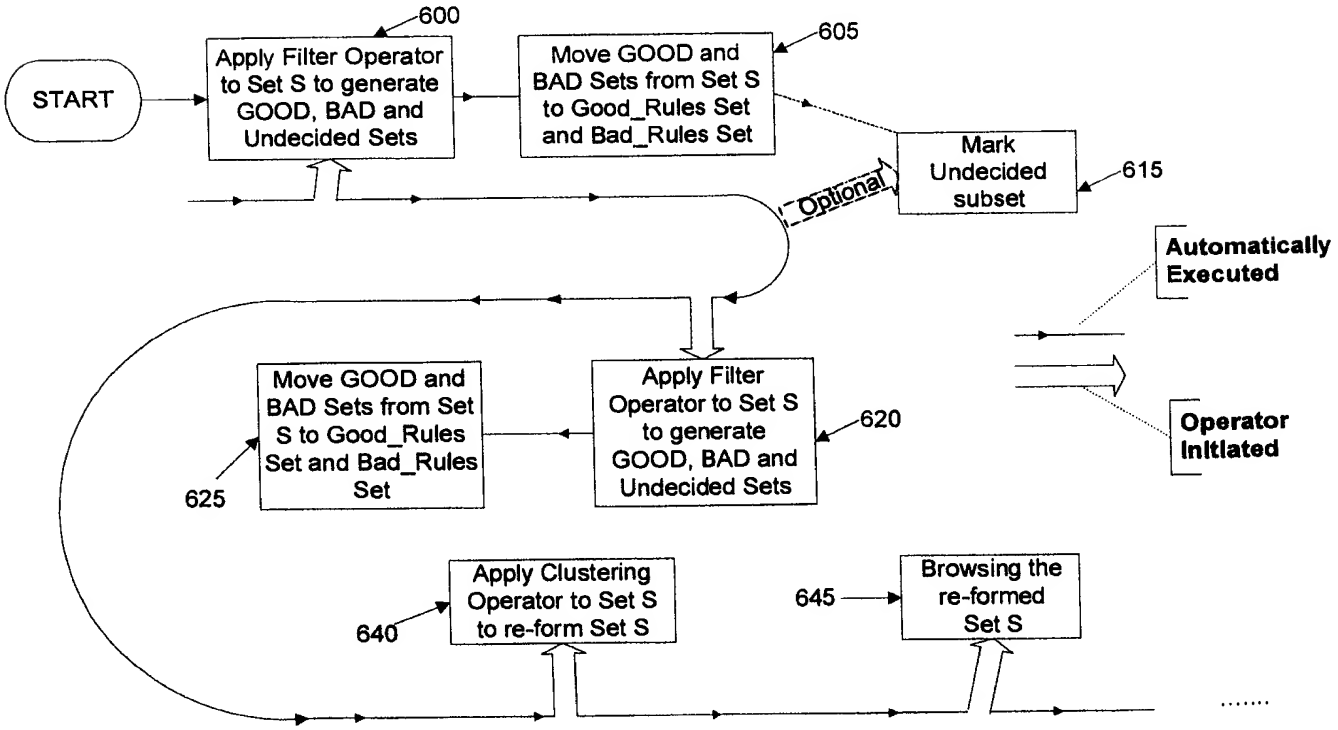


FIGURE 15

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/24339

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G06F 19/00, 17/30
US CL : 705/10, 26, 27
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 705/10, 26, 27; 707/1, 6; 379/189; 706/10; 345/357

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS, Dialog, IEL
Search Terms: profiling, profile, data mining, knowledge discovery, rule validation

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,P	US 5,727,129 A (BARRETT et al) 10 March 1998	1-32
X,P	US 5,727,199 A (CHEN et al) 10 March 1998 Figures 2 &4, col. 8, lines 16-51	1,3, 11, 15, 19, 26
X,P	US 5,790,645 A (FAWCETT et al) 4 Aug 1998, col. 3, lines 44-61. col 4, lines 4-42. col. 5, lines 55-60. col. 6, lines 35-40. Figures 2, 3, & 4	1, 2, 10, 11, 15, 19, 26
X	US 4,775,935 A (YOURICK) 4 OCT 1988 Col. 2, llines 39-68. Col. 3, lines 12-21. Col. 4, lines 8-27 & lines 63-68	1, 11-13, 15-22, 26, 30, 31
X	US 5,867,799 A (LANG et al) 2 FEB 1999 Figures 3 & 4, Col. 4, lines 46-60. Col. 5, lines 1-37. Col. 16, lines 22-44. Col. 30, lines 29-55	3, 4, 23

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 03 FEBRUARY 1999	Date of mailing of the international search report 08 MAR 1999
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer Joni Hill Allen MacDonald Telephone No. (703) 305-9708

Form PCT/ISA/210 (second sheet)(July 1992)★

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/24339

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	KOKKINAKI, A. I. 'On Atypical Database Transactions: Identification of Probable Frauds Using Maching Learning for User Profiling' In: Proceedings Knowledge and Data Engineering Exchange Workshop, 1997 . pages 107-113 All	1-75
A	CERQUIDES, J. 'Fuzzy Metaqueries for Guiding the Discovery Process in KKD' In: Proceedings of the Sixth IEEE International Conference on Fuzzy Systems 1997, vol. 3 pages 1555-1559. All	7
A	COOLEY et al. "Grouping Web Page References into Transactions for mining World Wide Web Browsing Patterns' Proceedings of the Knowledge and Data Engineering Exchange Workshop, 1997. pages 2-9 All	1-32